2011

Visualizing Biological Data in Google Earth

Ming Jia
Iowa State University

Visualizing biological data in Google Earth

by

Ming Jia

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Co-majors: Computer Engineering and Human Computer Interaction

Program of Study Committee:
Julie Dickerson, Major Professor
Suraj Kothari
Alex Stoytchev
Stephen Gilbert
Chris Harding

Iowa State University
Ames, Iowa
2011

TABLE OF CONTENTS

List of Figures

List of Tables

Abstract

Meaningful visualization of large scale biological data is the key for achieving new discoveries in system biology research. Typical types of biological data in research includes: biological pathways or networks, biological ontologies, and experimental data. Visualization tools used in these areas often fail to present a meaningful and insightful view of underlining data.

We present a new interactive visualization tool, MetNetGE, which features novel visualization techniques for three kinds of biological data: pathway, ontology and omics data. For a given biological pathway, we proposed a novel 3D layout algorithm, aligned 3D tiered layout, which arrange the pathway nodes into different tiers to make the cross-layer connection patterns stand out.

Biologists interested in a species may want to see all hundreds of metabolic pathways for that species. Instead of simply showing hundreds of pathways in one network in a complex and incomprehensible graph, MetNetGE organizes those pathways based on the hierarchical pathway ontology, and visualizes the structure using the proposed 3D Enhanced Radial Space-Filling (ERSF) technique. The ERSF algorithm uses an orbit metaphor to present the non-tree edges in the ontology. Mapping cumulative omics statistics on the ERSF drawing aids biologists in easily identifying highly activated pathways or categories in an experiment.

MetNetGE uses Google Earth (GE) as the underlining visualization tool. All the biological entities are converted to objects in the KML (Keyhole Markup Language) file and loaded in GE.

A user study with 20 participants to demonstrate the improved efficiency of MetNetGE over Cytoscape regards certain biological tasks. Although MetNetGE requires higher learning time (680 seconds vs. 350 seconds) on average, it helps participants quickly finish the tasks. Results showed that the completion time of using MetNetGE is about half of using Cytoscape.

Chapter 1. Introduction

Biomedical networks are widely studied to reveal the complex interactions between genes, gene products and cellular environments in biological processes [1-3]. Popular representations of such networks are the node-link graph and the adjacency matrix. In the node-link graph, nodes represent genes, gene products, metabolites and reactions, and edges represent specific interactions, e.g., transcription, translation, catalysis, and a variety of types of regulation.

The availability of high-throughput experimental data provides new possibilities to system biology, and creates new challenges for visualization tools as well. These experiments normally involve thousands of RNAs, metabolites and/or polypeptides. Mapping such data onto an interaction network can help biologists generate hypotheses about how the parts of the system influence each other.

A number of publicly accessible pathway databases are available, containing data about genes, gene products, and interactions, such as, BioCyc [2] MetNetDB [3] and KEGG [4]. In order to get better insight into such data sets, many visualization tools have been invented, e.g., Cytoscape [5], VisAnt [6]. In a review, Suderman et al. [7] studied 35 visualization tools and noted key useful features such as generation of good layouts and integration with analysis tools.

Despite the recent emergence of many pathway visualization tools, current tools are not suitable for many tasks. One important challenge is how to make visualization of the whole network meaningful. Other challenges include creating methods for showing hierarchical relationships, e.g., pathway ontologies, and devising new approaches for interactivity between analysis and visualization that may reveal hidden relationships.

These functionalities are crucial for biologists to explore, understand and make connections among the data.

Biologists wish to visualize the pathways organized in a meaningful manner. They also want an overview of the experimental values for the categories, e.g., they want to be able to ask questions such as whether degradation pathways have many genes highly expressed, or which categories are overrepresented in the data.

In order to meet these criteria, we have created a software platform, MetNetGE, which provides a hierarchical view of the pathway ontology and maps the experimental data onto this view. Preliminary user study with biologists in our group shows that MetNetGE can improve efficiency in many daily tasks and allow exploration of new patterns in the data.

MetNetGE is also designed to aid biologists in better understanding complex individual pathways, using a 3D tiered layout, where different entity types and interactions are located on different tiers. The algorithm computes layout based on the biologist's current selection of the most important plane, such that the pathway structure located on that plane stands out.

1.1. Background

Biological networks and pathways are widely studied to reveal the complex interactions between genes, gene products and cellular environments in biological processes [1-3]. Popular representations of such networks are the node-link graph and the adjacency matrix. In the node-link graph, nodes represent genes, gene products, metabolites and reactions, and edges represent specific interactions, e.g., transcription, translation, catalysis, and a variety of types of regulations. Fig 0.1 shows a typical pathway diagram in biology textbook [8].

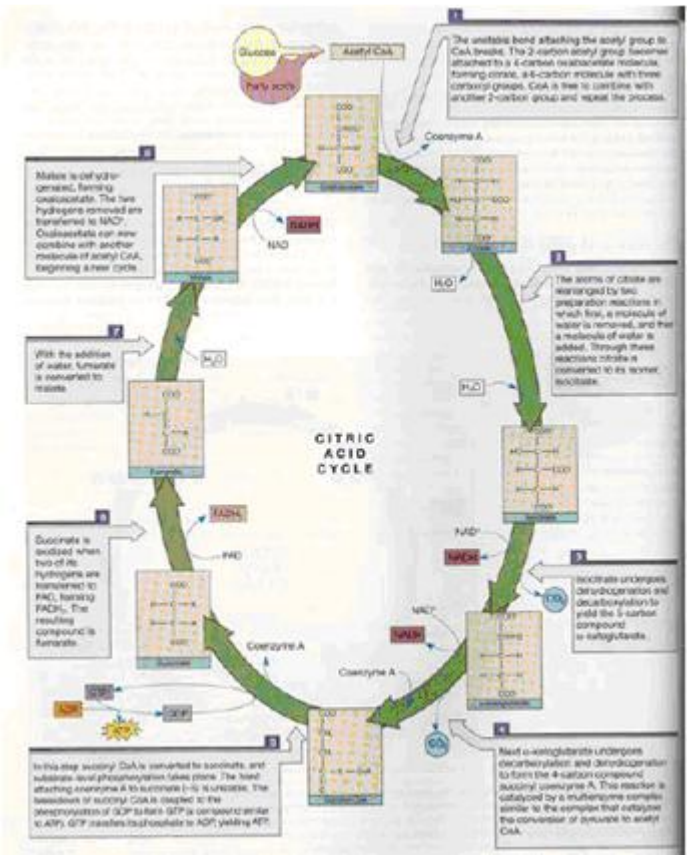


Fig 0.1 The pathway citric acid cycle shown in a biology textbook [8].

The availability of high-throughput experimental data provides new possibilities to system biology, and creates new challenges for visualization tools as well. These experiments normally involve thousands of RNAs, metabolites and/or polypeptides. Mapping such data onto an interaction network can help biologists generate hypotheses about how the parts of the system influence each other.

Biological Ontologies are part of an effort to create controlled vocabularies for shared use across different biological process. Typical ontologies are pathway ontology [9] and gene ontology [10]. Fig 0.2 shows the pathway ontology of *E.coli* in traditional tree list view from EcoCyc [11] web.

Fig 0.2 Pathway Ontology Tree from EcoCyc: http: ecocyc.org/

Understanding the PO structure helps biologists form a mental image of the interactions within the biological system. However, in day-to-day research, biologists need to make sense of system-wide experimental data and wish to understand how the experimental conditions affect the underlying biology. One typical type of experimental data is gene expression, which describes the abundance of gene transcripts during an experiment. Other experimental data include metabolomics and proteomics. For gene expression data, the original data is typically a data matrix where each row describes a gene, and each column records the expression level of genes under a certain condition, e.g., one time point, one treatment, or one replicate.

One pathway normally contains several genes, but it can range up to hundreds in signaling or regulatory pathways. One pathway category, in turn, contains several pathways and by extension a group of genes. Therefore, we can define a group of genes for each pathway and category. In order to understand the experiment on a functional or even system-wide level, biologists try to derive aggregated values for each pathways and

categories, e.g., the average expression for all the genes in a pathway or the number of differentially expressed genes and their p-values.

1.2. Existing Solutions and Limitations

We visualized three different kinds of biological data: the pathway diagrams, pathway ontology, and the omics data. The data which is most well studied is the pathway diagram. Dozens of visualization tools were developed and several of them are still under active development nowadays. In a review, Suderman et al. [7] studied 35 visualization tools and noted key useful features such as generation of good layouts and integration with analysis tools.

One of the most important features for the pathway visualization tool is how good the tool is to layout the pathway in a meaningful manner. The traditional 2D based layouts are very suitable for small pathways, e.g. the pathway with less than 30 nodes, and 50 edges. However, when the pathways become larger and larger, edge crossings will occur very frequently, and the graph's connectivity will be difficult to determine.

3D algorithms can eliminate the edge crossings by arranging nodes in 3D spaces. However, since the look of the pathway may change too much from its 2D counterpart, biologists who are used to viewing the pathway in 2D have difficulty understanding the 3D diagrams. Another problem of 3D layout algorithm is that it can be quite hard to navigate a network in a 3D space, especially when using input devices, such as keyboard and mouse [12].

Biologists often view and explore the ontology on websites, e.g. ecocyc.org, or geneontology.org. All these websites use the tree list view to present the hierarchical structure of the ontology. Some desktop tools also exist to aid the exploration of complex ontology using node-link graphs, e.g. OBOEdit [13]. However, the tree list view and

node-link graph have limitations that they can't view the whole ontology structure in a single computer screen, thus failed to generate an overview of the ontology. Moreover, the ontologies are actually directly acyclic graphs (DAG), where each node may have multiple parents. All the above methods can only show tree structure and duplicate the nodes that have multiple parents, thus the important existence of multiple inheritance is hidden. Another limitation of those methods is that they can not map multiple attributes onto the ontology nodes.

One of the most well studied and important experimental data is transcriptomics data, also known as gene expression data. It recorded the expression level of each gene under certain experiment conditions. Typically, researchers will store and view this data in a large data matrix, where each row represents one gene and each column represents one condition. Since combining the transcriptomics data with the pathway diagram may lead to new discovery about the pathway, many visualization tools provide the ability to map the expression values on the pathway diagram. However, since normal pathway layout algorithms can not handle very large graph, the number of genes they can show at a given condition is also limited. This limitation prevents biologists from studying the transcriptomics data in a large and system level scale.

1.3. Overview of Proposed Methods

Based on the survey [7] in biological visualization field and interviews with our biologist collaborators, the requirements for the pathway visualization are:

- Reduce edge crossings to maintain a comprehensible with of the pathway structure which may contain more than 50 nodes.
- Clearly show the pathway structure
- Make the significant part of the pathway stand out.

- Showing detailed information on demand, i.e. hide unwanted information.

The requirements for ontology and experimental data visualizations are:

- View the whole ontology in one screen to have a global feeling of the data and the main hierarchical structure.
- View details by navigation and/or interaction (zoom, pan, rotation).
- Map experimental data and other aggregated attributes on the ontology so that they are easily visible.
- Clearly show non-tree connections.

During our research, we developed the software platform MetNetGE, which can provide an integrated visualization solution for all above three types of data: the pathway diagrams, pathway ontology, and the omics data.

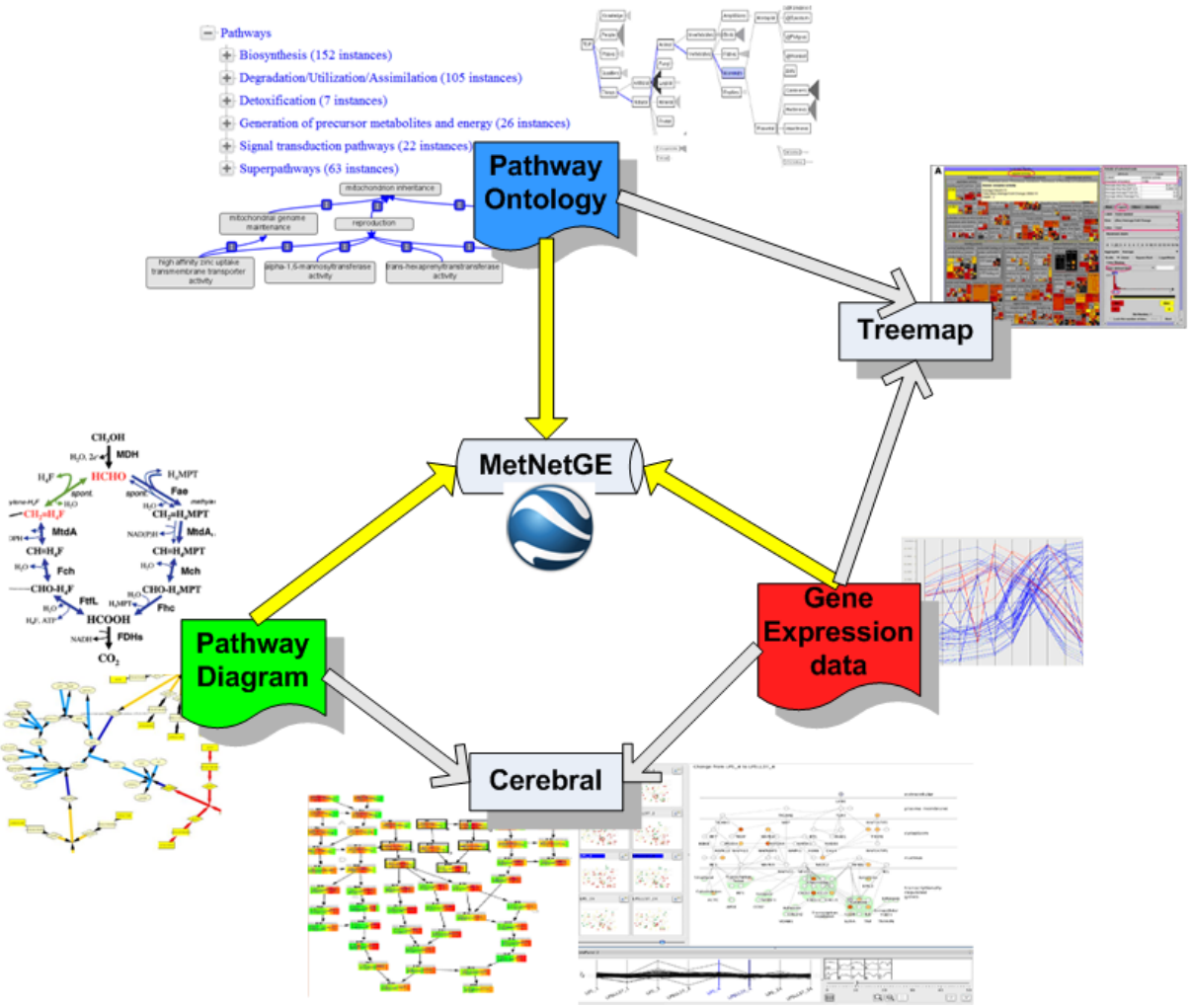


Fig 0.3 The three types of data visualized in MetNetGE.
The small picture around each data is some traditional methods to view them.

Fig 0.3 shows the three different types of data which can be visualized in MetNetGE. The small pictures around each data are the traditional methods to view them. There are also several existing approaches to integrate two of the three types of data. For example, Cerebral [14] and many other tools maps gene expression data on pathway diagram, and Baehrecke et al. [15] use treemap to map the expression data on gene ontology. MetNetGE is the first system which can integrate all the above data into one visualization system.

Since there may exist hundreds of pathways for a given species, laying out all of them together inevitably will result in a dense and cluttered graph. We try to avoid this

problem by organizing the pathway diagrams by pathway ontology. We proposed our algorithm, Enhanced Radial Space-Filling (ERSF) technique to layout and show ontology. Each ontology node is represented as a colorful region in the drawing, and the detailed pathway diagram is drawn inside the region.

To handle larger and much complex pathway diagram, we proposed the novel Aligned 3D Tiered (A3T) layout algorithm, where the graph complexity is reduced by separating the nodes into 4 parallel planes and were aligned in 3D space. By controlling the transparency of each plane, we can easily hide the unwanted graph details and show them when user needed.

To enable the study of large scale transcriptomics data, we map some cumulative expression value on the pathway ontology drawing. As a result, biologists can easily identify which pathways and categories are highly expressed in a certain condition. Moreover, we can also map the difference of expression values between two conditions, thus help biologists in finding differentially expressed genes. We also provide the ability to visually navigate the gene expression data on individual pathway by drawing extruded polygons on the pathway diagrams. The parallel coordinate plots are also one helpful way to see the trend of many genes across several conditions.

In order to demonstrate the improved efficiency of MetNetGE, we conducted a user study with 20 participants. Participants used MetNetGE and the comparing tool Cytoscape to finish selected biological tasks after completing a tutorial section. The tasks are selected based on the visualization requirements and are the abstractions of real tasks biologists need to perform in day-to-day work. Although MetNetGE requires higher learning time (680 seconds vs. 350 seconds) on average, it helps participants quickly finish the tasks. Results showed that the completion time of using MetNetGE is about half

of using Cytoscape. For example, to find highly related categories in one pathway ontology of about 200 nodes, participants averagely used 133 seconds in MetNetGE and 324 seconds in Cytoscape.

1.4. PhD Research Contributions

My PhD research includes contributions to engineering, algorithm and human computer interaction (HCI) areas. From the software engineering perspective, I developed the software platform MetNetGE to support 3D visualization of biological data in Google Earth (GE) and the contributions include following:

- Provided an integrated visualization solution for pathway, ontology, and omics data.
- Developed APIs for python software to create 3D geometries, icons and animations in GE.
- Developed many interaction methods with GE.

From the algorithmic perspective, my contributions are following:

- Proposed and implemented the novel Aligned 3D Tiered (A3T) layout.
- Proposed and implemented the novel 3D Enhanced Radial Space Filling (ERSF) layout.
- ERSF does not replicate nodes when the multiple inheritances exist in the heretical structure, which reduces graph complexity by 13% (for pathway ontology) to 26% (for gene ontology).
- Link the traditional treelist view with the RSF drawing, to enable detailed navigation with context visualization.

- Proposed and implemented various ways to map gene expression data and over representation value on the ontology visualization to enable the study of omics data on both pathway and gene ontology.

From the HCI perspective, my contributions are following:

- Conducted a relatively large scale of user study including 20 participants to evaluate MetNetGE and Cytoscape.
- Used statistical methods to analyze the study results.
- Used HCI related theories to propose possible explanations of the user study results.

1.5. Organization

This paper is organized as follows: Chapter 2 describes details about the related work and current solutions. Chapter 3 shows the implementation and framework design of MetNetGE. Chapter 4 illustrates the proposed 3D Tiered layout, comparing it with existing layout algorithms. Chapter 5 introduces the use of 3D ERSF technique to visualize and interact with ontologies. Chapter 6 explores the various ways to map experimental data on the ERSF drawing. Chapter 7 presents some preliminary result with several typical working scenarios of MetNetGE. Chapter 8 uses the journal paper format to include one paper prepared for submission. Finally, Chapter 9 concludes the whole thesis.

Chapter 2. Related Work

2.1. Biological Pathway Visualization

A number of publicly accessible pathway databases are available, containing data about genes, gene products, and interactions, such as, BioCyc [2] MetNetDB [3] and KEGG [16]. In order to get better insight into such data sets, many visualization tools have been invented, e.g., Cytoscape [5], VisAnt [6]. In a review, Suderman et al. [7] studied 35 visualization tools and noted key useful features such as generation of good layouts and integration with analysis tools.

Despite the development of many pathway visualization tools, current tools are not suitable for many tasks. One important challenge is how to make visualization of large networks meaningful. Unfortunately, the current tools with traditional layout algorithms typically results in the notorious ‘hair-ball’ view (as in Fig 2.1) for such a densely connected network. This view gives the user very little information about the structure of the network. Other attempts to make meaning out of large graphs include color coding according to feature, layouts that separates out parts of the graph by features, etc. Other challenges include creating methods for showing hierarchical relationships, e.g., pathway ontologies, and devising new approaches for interactivity between analysis and visualization that may reveal hidden relationships. These functionalities are crucial for biologists to explore, understand and make connections among the data.

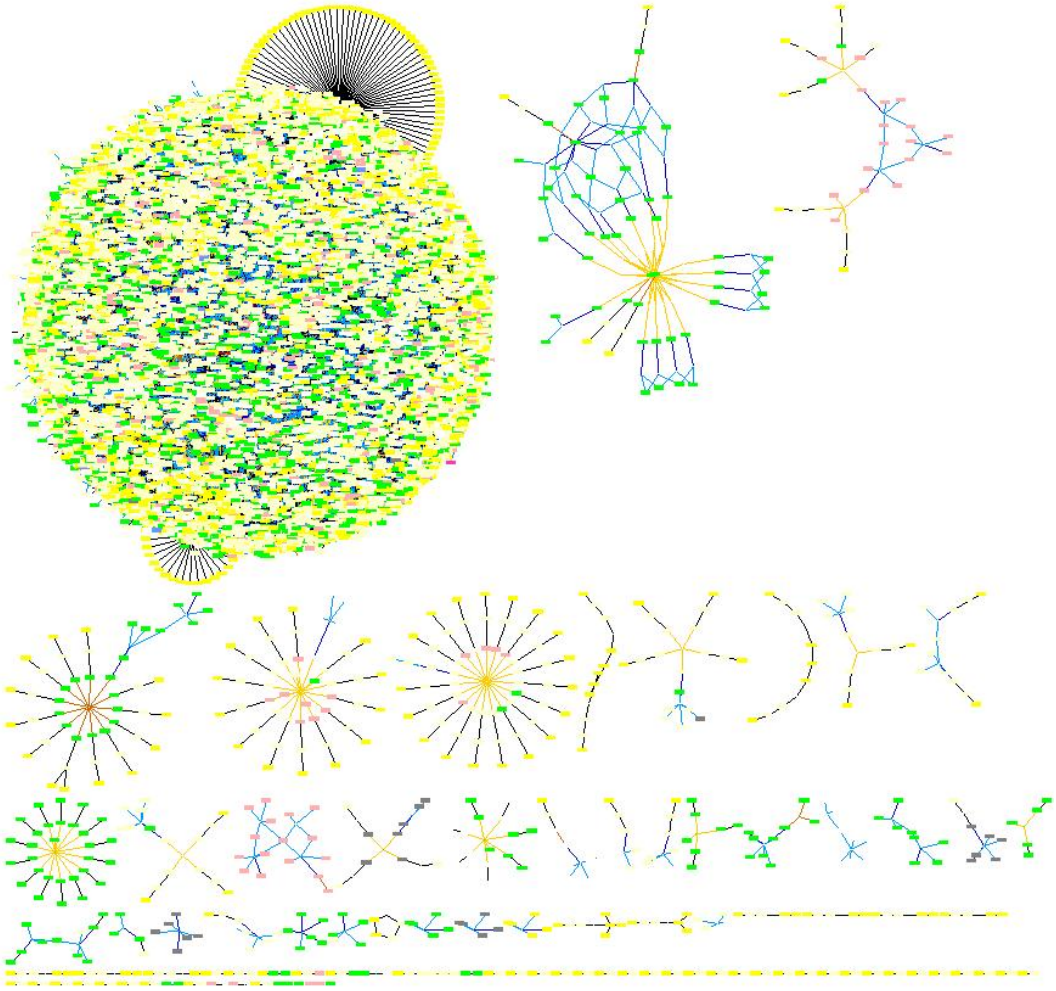


Fig 2.1 All the pathways of Arabidopsis from MetNetDB are loaded in Cytoscape and shown in organic layout.

2D layouts are especially favorable for the visualization of individual pathways since traditional diagrams are all drawn in 2D. 3D approaches exist [17, 18], however most popular bioinformatics tools do not support 3D directly. Some reasons that 3D layout methods are not widely adopted are: biologists are used to 2D representation, and it's hard for them to make sense of those 3D structures and 3D spaces can be hard to navigate. Even with proper navigation tool such as 3D mouse, the advanced users still need a lot of time to smoothly explore the space [12]. Fig 2.2 and Fig 2.3 shows the pictures of some 3D approach.

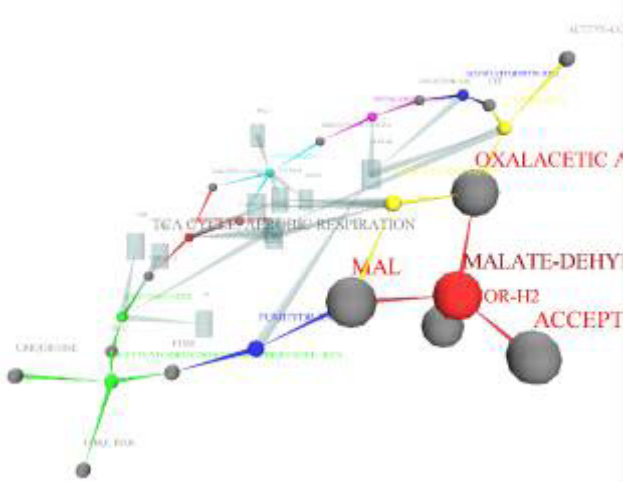


Fig 2.2 A snapshot of VRML Metabolic Network Visualizer [17].

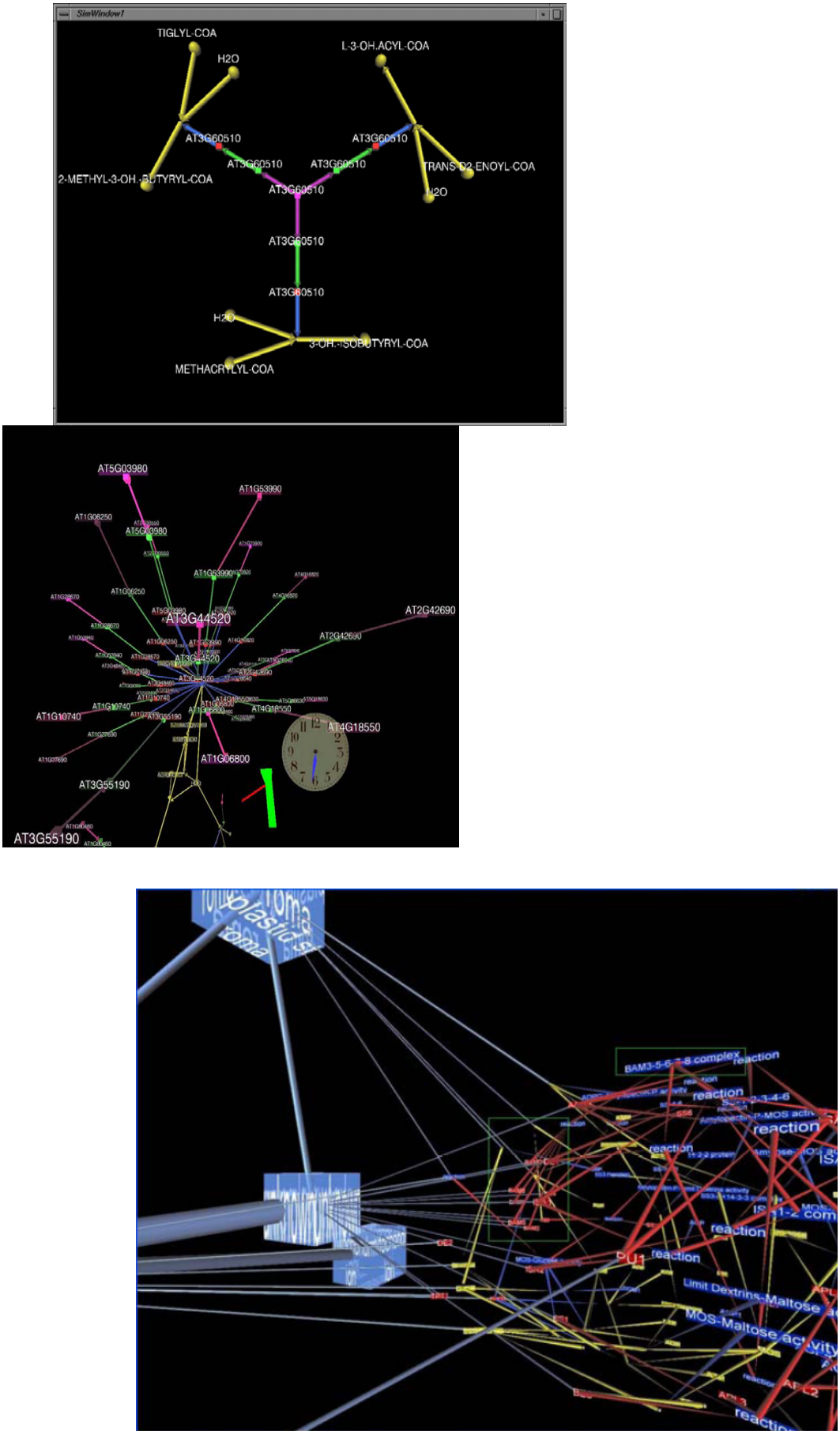


Fig 2.3 Metabolic pathways are shown in MetNetVR [18] in virtual reality.

The use of stacked 2D layouts was introduced in [19], where similar pathways across several species are compared. This representation (as shown in Fig 2.4) is very effective at highlighting small differences between two species; however it is not suitable to be directly applied to individual pathway diagrams due to the lack of connections between layers.

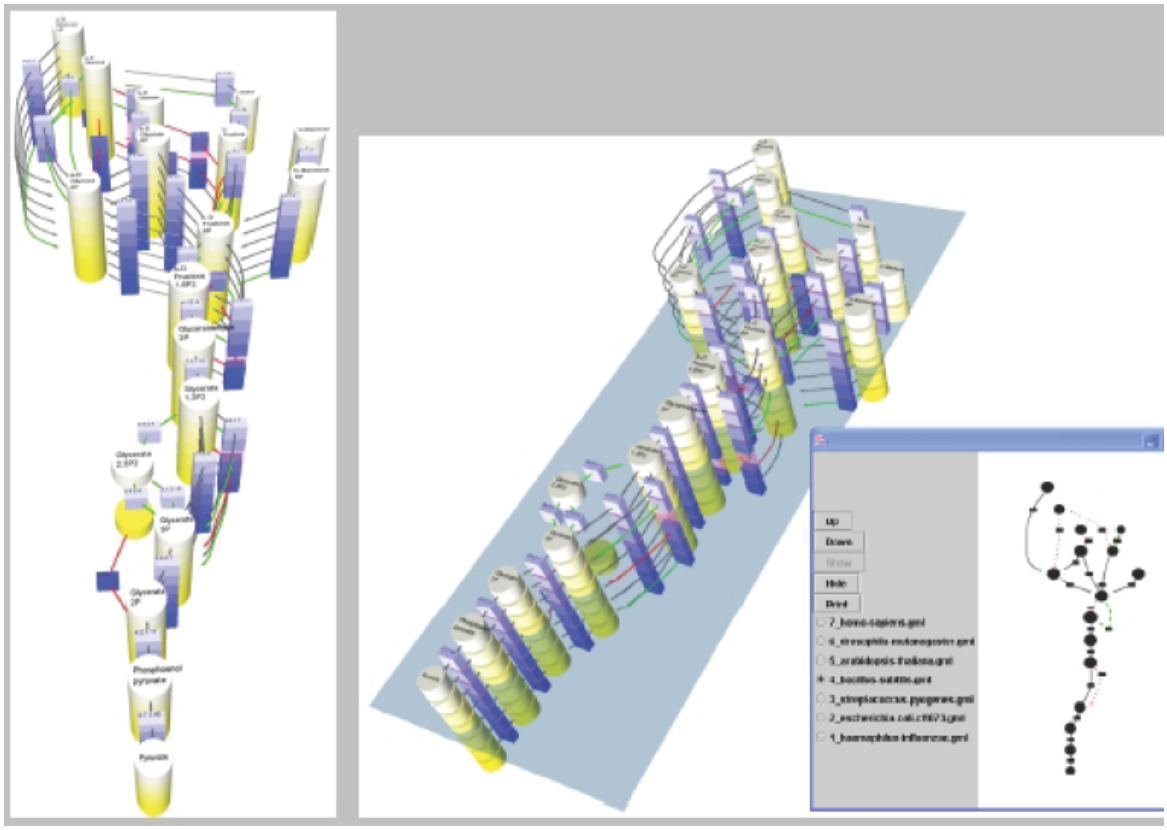


Fig 2.4 Visualizing related metabolic pathways in two and a half dimensions [19].

Arena3D [20] puts nodes into different layers based on node type to reveal several cross layer insight (in Fig 2.5). BioCichlid [21] divides protein and genes into separate layers to visualize the cross layer patterns. These works show the promise of using an extra dimension where the network complexity is reduced by separating the whole graph into several 2D planes. However, since they compute separate layouts for each layer, edges between layers are often cluttered and difficult to follow when used in individual pathways. We independently proposed to utilize the 3D tiered layout for each individual

pathway using an aligned 3D layout. The algorithm aligns the layers based on a user selected important plane to make the basic pathway structure stand out and create more aesthetic drawing.

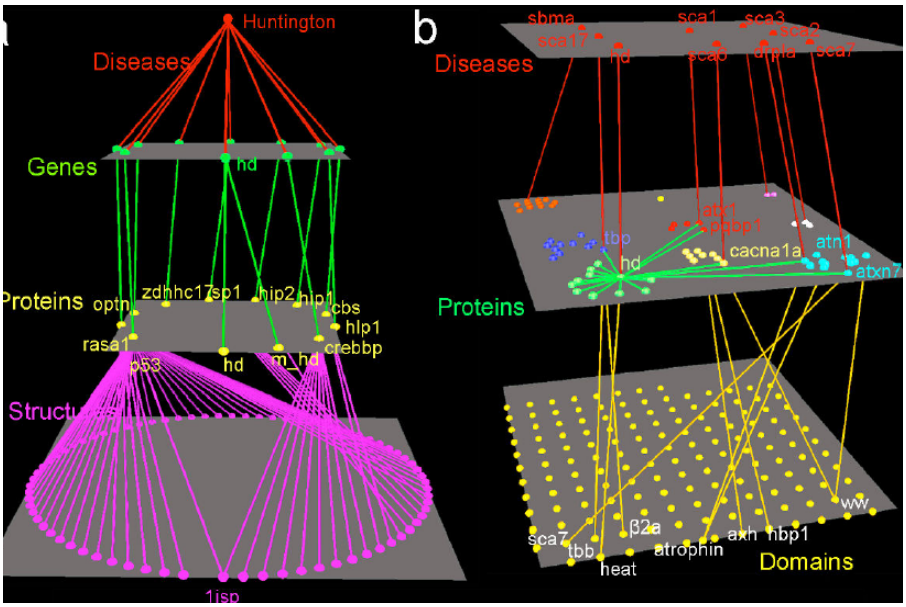


Fig 2.5 The layered layout in Arena3D [20] to show relationship between proteins and genes.

When biologists want to view the whole picture of all pathways in a species, they normally using visualization tool to load all pathway data, and draw them in the same view. However, due to the large number of pathways, there may be thousands of nodes in the view and most of them are connected, which results in a hair-ball like structure (Fig 2.1). The dense ball on the upper left part in the view is impossible to interpret even when zoomed in and using color coding on the network.

Researchers have tried ways to better organize all pathways. MetaViz [22] can preserve the structure of important large pathways, and show components in superpathways in adjacent regions (in Fig 2.6). Although the superpathways do keep their structures, the rectangular black lines connecting pathways are still difficult to trace. User

studies found that the rectangular layout is quite ineffective for tasks regarding understand the topological structures [23].

Instead of keeping the connections between pathways and generating an incomprehensible graph, EcoCyc [11] employs a cellular overview for all *E.coli* pathways where each pathway is represented as a small diagram (in Fig 2.7). The grey regions wrapping the pathways indicate that those pathways are from the same categories under the notation of pathway ontology.

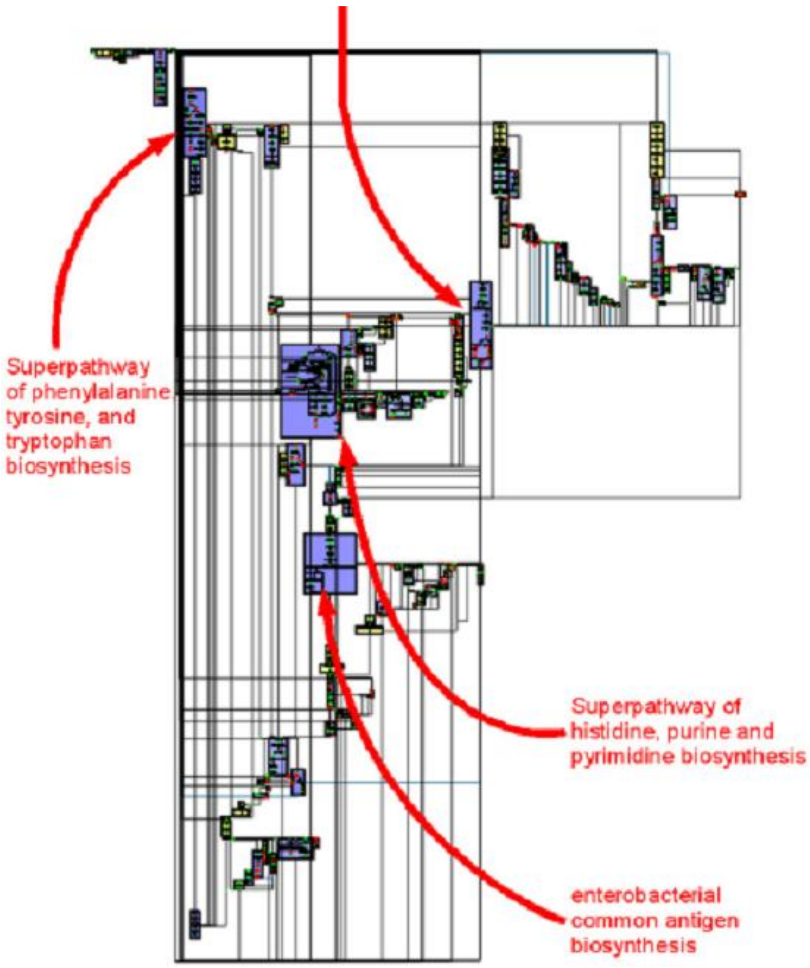


Fig 2.6 Whole metabolic network of E. coli drawn by MetaViz [22].

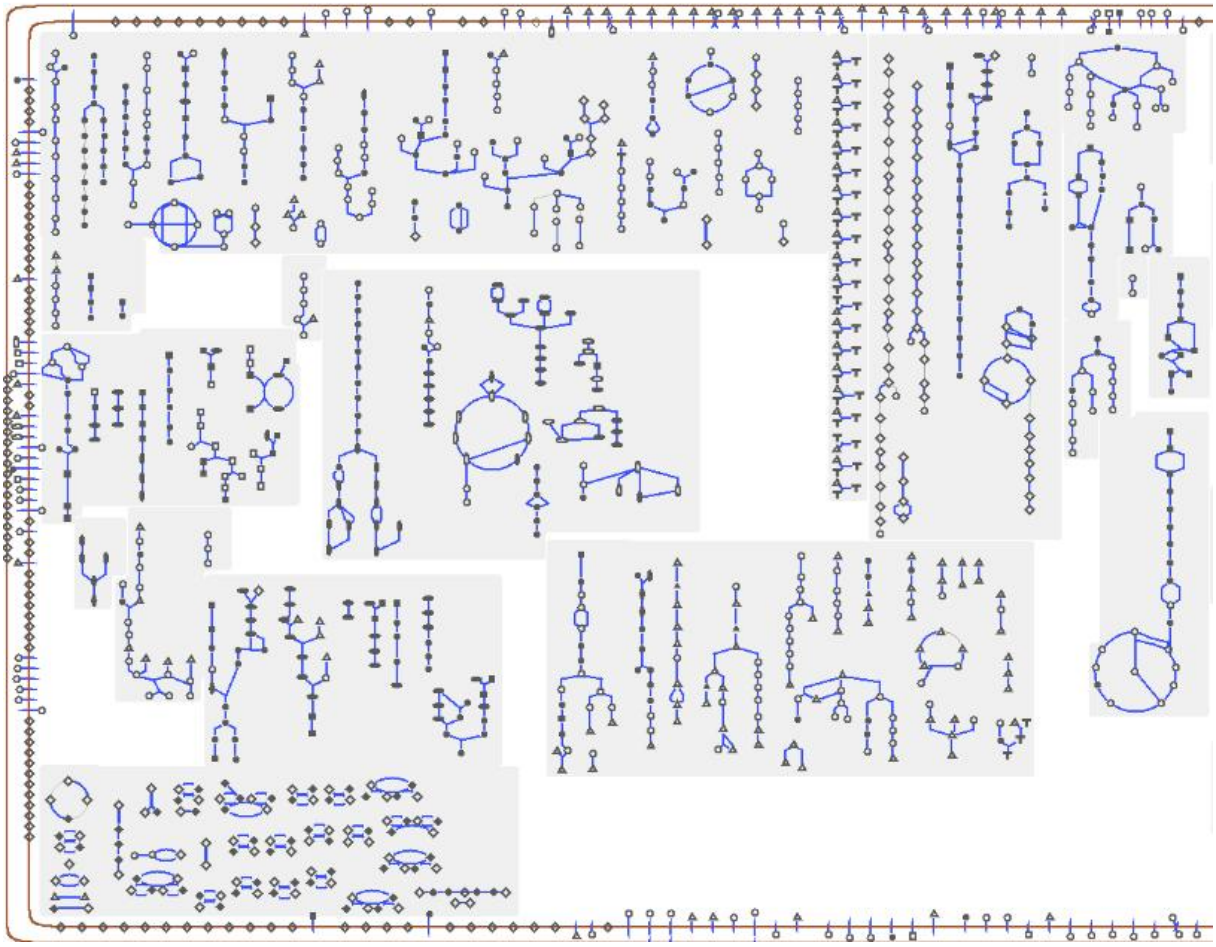


Fig 2.7 Overview of the Escherichia coli K-12 substr. MG1655 Metabolic Map from EcoCyc [11].

2.2. Biological Ontology Visualization

As shown in the EcoCyc omics viewer, structures such as ontologies can help organize biological information. Therefore we organize biological network by pathway ontology (in Fig 0.2), which consists of a spanning tree and several non-tree edges. Tree visualization is a well-studied research topic: various techniques have been proposed and implemented to support trees containing thousands of nodes. Popular methods are treemap [24, 25], radial space-filling [26], cone tree [27], and hyperbolic layout [28]. Hyperbolic trees are suitable in exploring large tree or near-tree structures. However, since we want to insert detailed pathway structures into the leaf nodes, we cannot use the hollow sphere employed in such hyperbolic trees. Also, the small space used to draw each node is not appropriate for mapping experimental data. Treemap (Fig 2.13) has the

advantage to show attributes for thousands of leaf nodes, however, it is not suitable to show attributes for non-leaf nodes (which may consist half of the nodes in biological ontology). Moreover, general treemaps can not show non-tree edges which are very common in biological ontologies. Fekete [29] tried to overlay the non-tree edges on a treemap (in Fig 2.8), however, this attempt creates many edge-crossings which makes the task of tracing non-tree edges difficult.

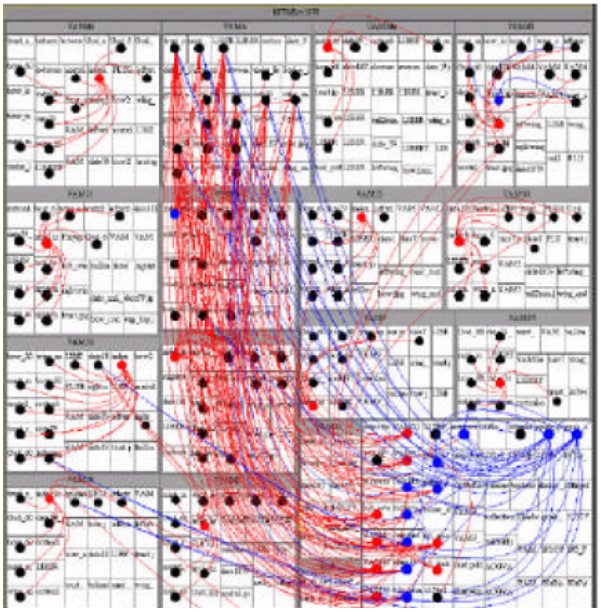


Fig 2.8 Overlaying non-tree edges onto treemap [29].

Current tools to visualize biological ontology normally use the traditional windows-explorer-like tree list, e.g. EcoCyc [11] and AmiGO [30]. Some desktop applications designed specifically to show ontology are also available, e.g. OBOEdit [13] and BinGO [31] (in Fig 2.10). The OBO-edit (in Fig 2.9) can visualize the ontology with both one windows explorer-like tree browser (Tree Editor) and one graphical tree drawing (Graph Editor). However, these tools all utilize node-link based top-down hierarchical layout to graphically represent the ontology. This kind of layout, such as dot, is well suited for dozens of nodes, however, will quickly become cluttered if all hundreds of ontology nodes are shown together. As a result, users of these tools normally collapse the whole

ontology, and only expend the hierarchy to the required extend, thus lose the context of the whole ontology structure. Moreover, biological ontologies are not pure tree structure, but the Directed Acyclic Graph (DAG), i.e., several child nodes have multiple parents. Current tools are not suitable to trace such connections.

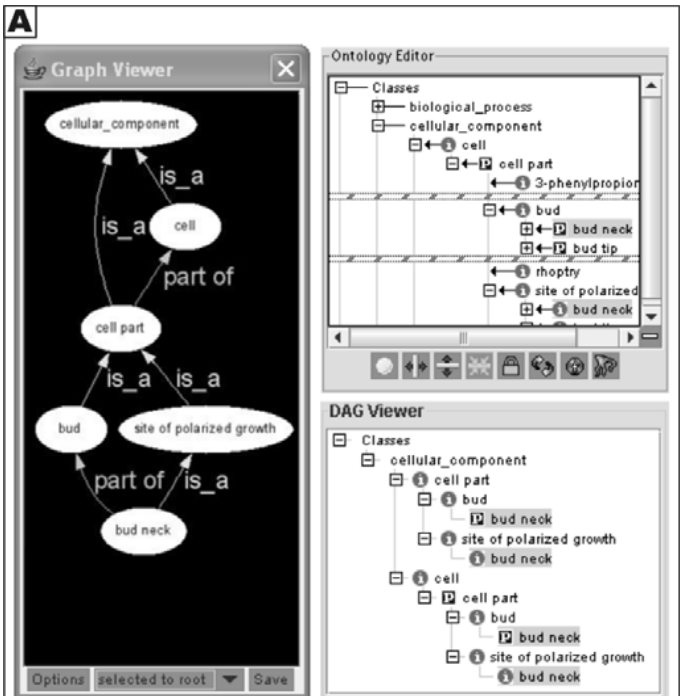


Fig 2.9 OBOEdit [13] which can view and edit gene ontology structure.

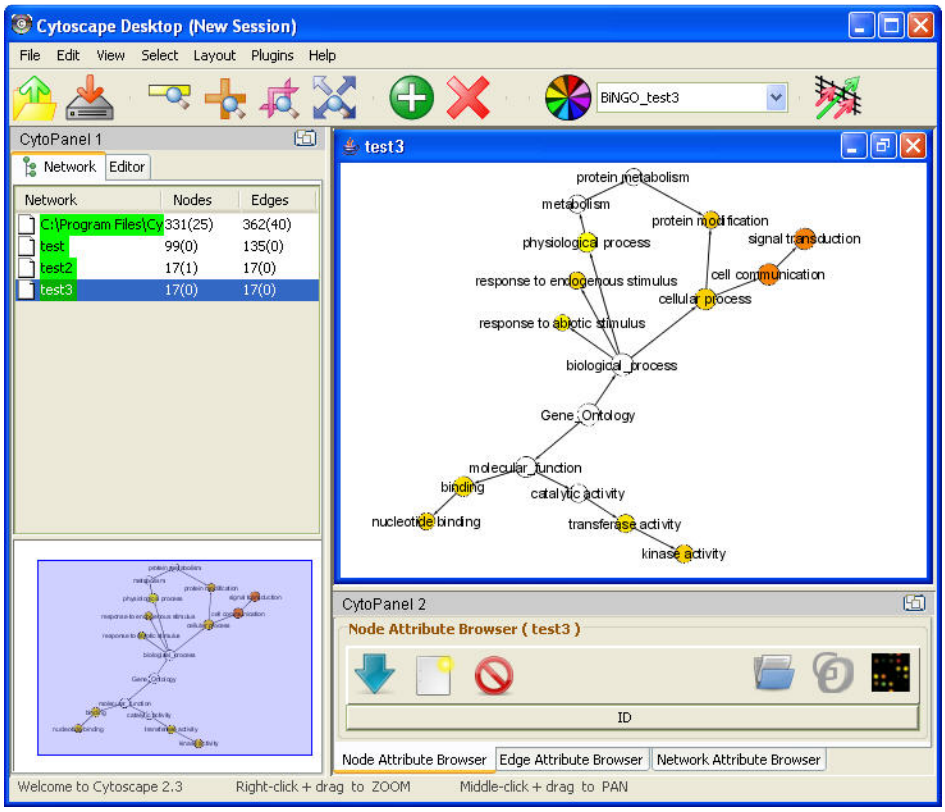


Fig 2.10 The Cytoscape plug-in BinGO [31], which can view gene ontology inside Cytoscape.

As mentioned previously, all the tree visualization method can only handle pure tree structures. If the input graph is a DAG, the above methods duplicate nodes. The node duplication not only increases the graph complexity, but also can not review the interesting multiple inheritance information. If users need to map some attribute on the graph, the result gets even more confusing, e.g. one drawing may review that two different parts in the graph are highly active, but finally find out they are only duplicated nodes of each other.

2.3. Transcriptomics Data Visualization

Transcriptomics or gene expression data recorded the expression level of each gene under certain experiment condition. Typically, researchers will store and view this data in a large spreadsheet, where each row represents one gene and each column represents one condition. Besides organizing the pathways in a meaningful manner, biologists also want an overview of the experimental values for the categories, e.g., they want to be able to ask

questions such as whether degradation pathways have many genes highly expressed, or which categories are overrepresented in the data.

Many works have been done to incorporate gene expression data onto pathway diagrams. Cerebral [14] is a Cytoscape plug-in which allows user to map gene expression data on loaded pathways, and mapping the absolute value as well as difference between conditions on node colors (in Fig 2.11). Researches also compared many representations of time series gene expression data [32], e.g. heatmap, line charts and complex node glyphs (in Fig 2.12).

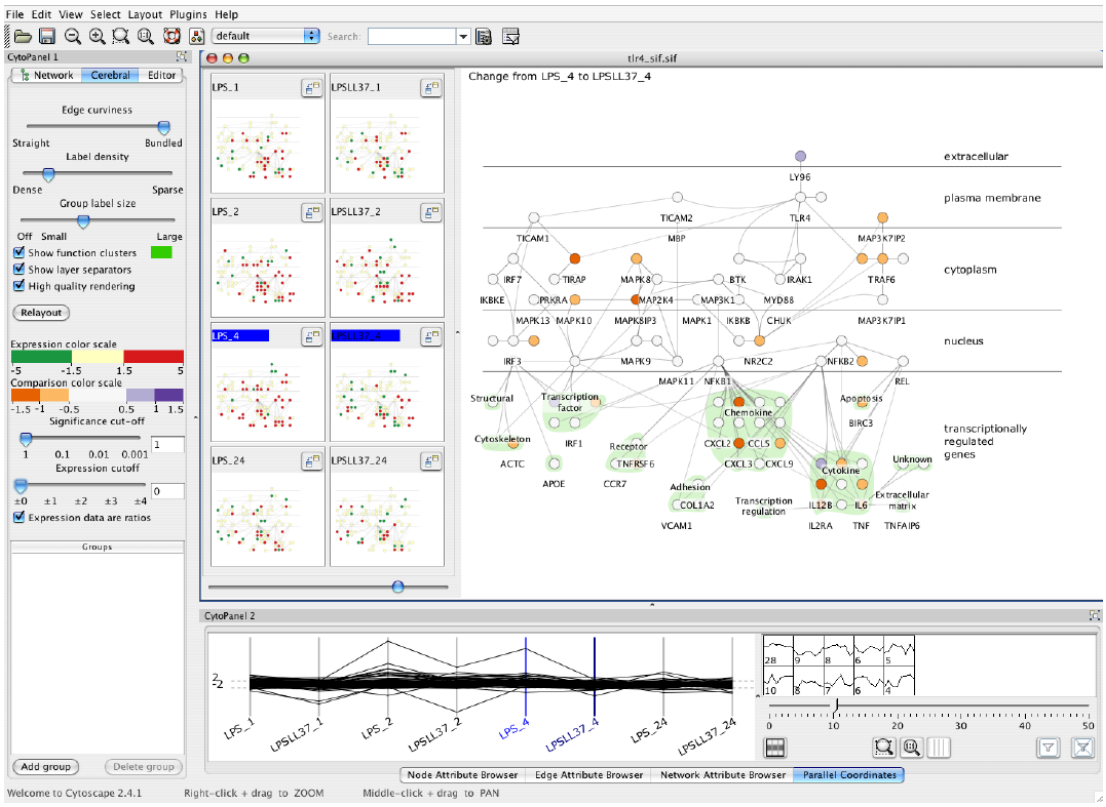


Fig 2.11 Gene expression data are mapped onto pathway diagram in Cerebral [14].

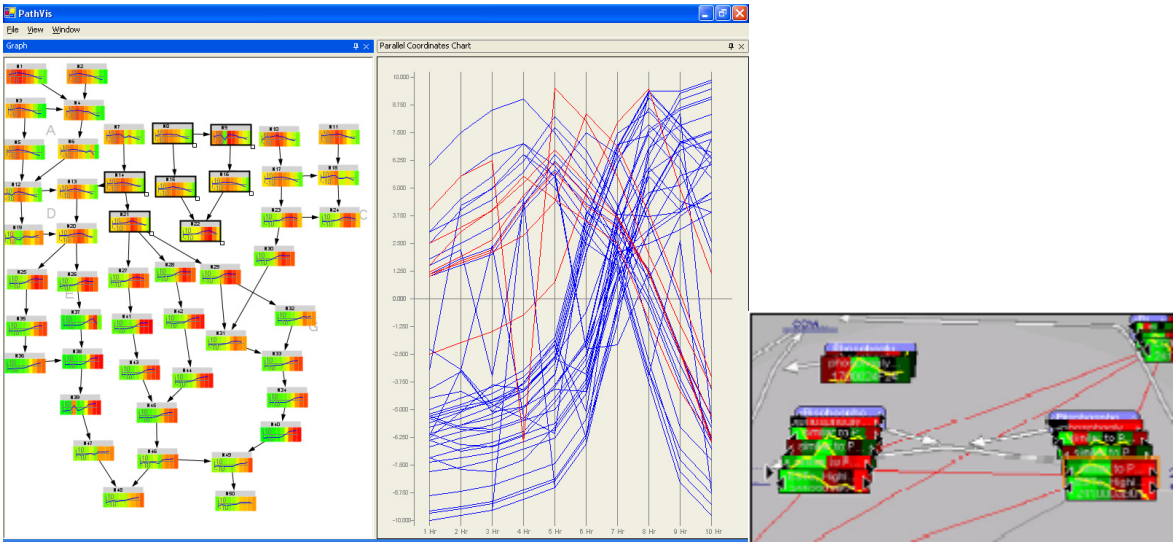


Fig 2.12 Comparison of different strategies to map gene expression data on pathway nodes [32].

Gene expression data can also be mapped to gene ontology. Baehrecke et al. [15] have mapped microarray gene expression data on gene ontology with treemap. Their representation is good at showing the information for bottom level ontology terms; however, the hierarchy of ontology is not clear. Since the gene ontology is not a pure tree, many ontology terms present in several different regions, which makes the drawing a little bit confusing, e.g. user identify two regions have same expression value, but only to find out they are actually the same region.

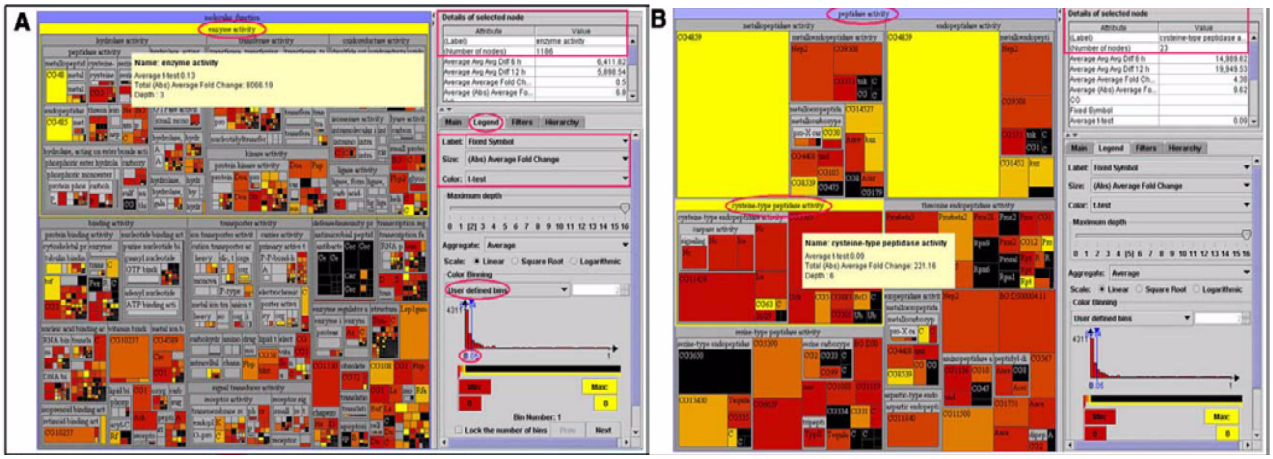


Fig 2.13 Use treemap to show gene ontology and microarray data, duplicating gene nodes since it's not a tree.

Chapter 3. System Framework and Implementation

3.1. System Framework

The MetNetGE system is composed of four major modules: pathway loader, ontology loader, Transcriptomics loader, and Graphical User Interface (GUI). Fig 3.1 shows the framework and the relations between those major modules. Users are mainly interacting with the GUI to open files, and customize options. User can also directly navigate the drawing in GE using GE's own navigation widget; however, the navigation assistance from MetNetGE is very helpful in exploring our dataset.

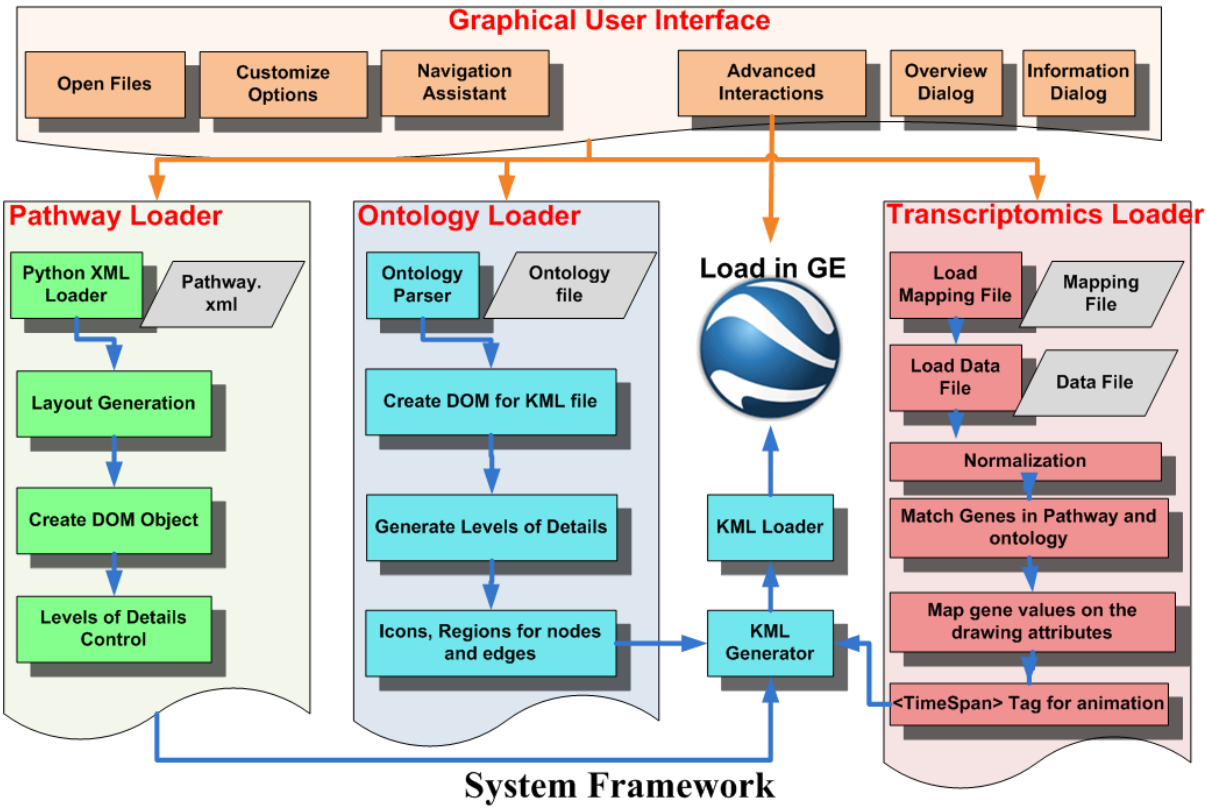


Fig 3.1 System framework of MetNetGE.

The system contains four major modules. The loader modules are responsible for different data types. The loaders process the data, generate layouts, and map the data to KML elements. All data are generated as KML file and loaded in GE.

A typical working scenario of using pathway loader is that a user first opens one or multiple pathway files through GUI, then customizes the setting of how he/she wants the

network to be generated, such as layout, color and icon style, then generate the KML file for pathways and finally visualize the file in Google Earth (GE).

The use of ontology loader is quite similar, and we support the loading of general ontology data in obo format.

After loading either the pathways or gene ontology, user can use experimental data loader to read a data file and mapping file. In the current stage, we support the gene expression data in plain text format, e.g. comma-separated values (CSV), or tab separated values. The program will then use the mapping file to scan the pathway or ontology data, and finally associate the data with loaded pathways or ontology. Then user can choose different visual mapping options, and generate the KML file to show in GE.

The full User Interface of MetNetGE consists of three dialogs which are all floating on top of Google Earth window. The main dialog (Fig 3.2a) contains all the functions to import, customize and create the ontologies and pathways. User can choose the 3D tiered layout, or change to dot, spring or many other layouts. User can also make the planes of each tier hidden or transparent to expose the underlining structures.

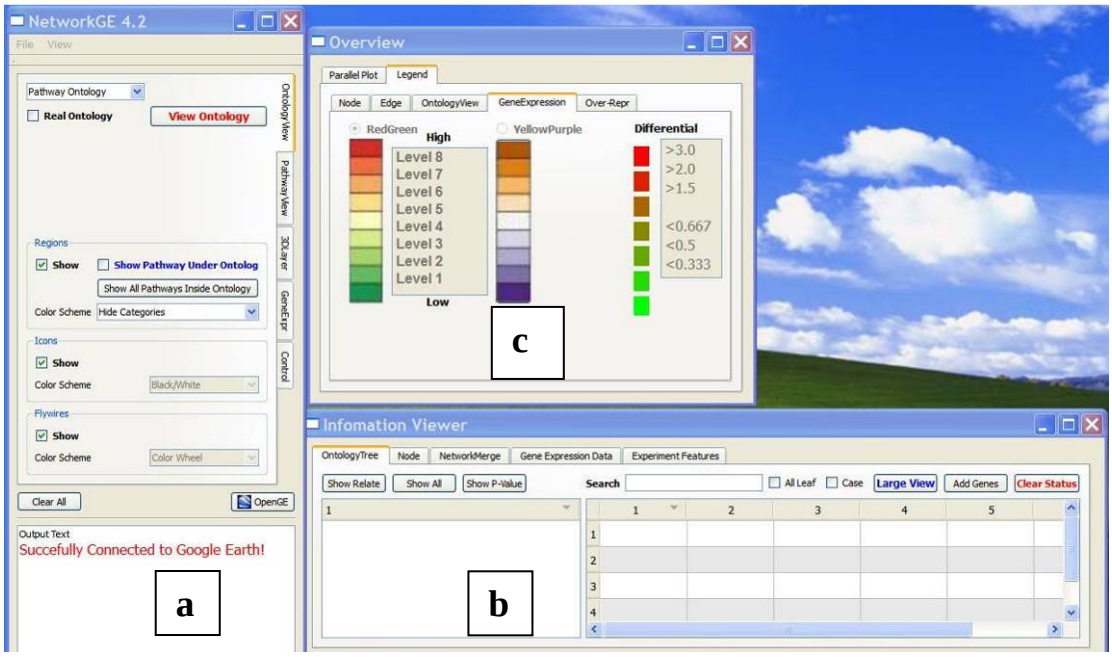


Fig 3.2 The dialogs of MetNetGE's GUI.

(a) Main dialog contains buttons for loading data, navigation and customization of the visualization. (b) Information dialog contains detailed information of ontology, pathway and gene data. (c) Overview dialog contains parallel coordinate plot and legends.

The information dialog (Fig 3.2b) contains a traditional list and table view of the loaded ontology, which is linked to the ERSF drawing in Google Earth. The selection of ontology item in the list/table will highlight the corresponding region in the ERSF drawing and vice versa.

The graphics dialog (Fig 3.2c) can show the parallel coordinate plots of gene expression value for the selected genes. The poly-line which represents the currently selected gene will be highlighted as red. This dialog also contains the legends for all the information drawn in Google Earth.

In addition to the use of orbits metaphor, MetNetGE provides some interactions to make the task of viewing relationships between ontology regions quite simple. User can first select one region by clicking on either the ERSF drawing or the ontology list/table in information dialog. Then, by pressing the button "View Relate", all the ontology regions will be colored differently based on their relation to the selected region, e.g. regions share

a common child with the selected ontology node will be colored dark yellow. Furthermore, if user clicks on one access point, he will be provided with options to view the child or the parent.

MetNetGE also provide several functions to aid the navigation in 3D space, e.g. user can click buttons to fly to the corresponding tier of selected pathway. Interested users can view the documentation on project website (www.metnetge.org) to know more.

3.2. Implementation

MetNetGE was implemented in Python. After the loading and computation, all pathway and ontology drawings were created as KML (Keyhole Markup Language) files, and were loaded into Google Earth through its COM API. The graphical user interface (GUI) is written with PyQt. To run MetNetGE, user need to install Python 2.5 or above, and several dependent open source python libraries. The documentation on project website (www.metnetge.org) provides download for all required libraries.

3.3. Icon Representation

We use icons to represent entity nodes in the pathway. Using icons have many advantages over traditional use of color and shape combination for three reasons. First, human perception can only easily distinguish roughly one dozen color hues [7], which are not enough for a variety of information biological data presented. However, the number of distinguishable icons is much more than a dozen. Second, icons in GE are shown with fixed screen size, i.e. no matter you zoom in or out, it always occupies a fixed area convenient for recognition. This characteristic is very helpful when a viewer zoom out to see the larger structure while still want a clear sight of the names of nodes. Third, icons in GE are clickable, and its description tag would be shown as a dialog when it is clicked. In this way, we can store all the detailed information of each node in the description tag of icon including URLs.

3.4. Control of Levels of Detail

Since the metabolic network normally contains many pathways, showing them all together would challenge user's cognitive load capacity. To address this problem, we have explored one of GE's formal mechanisms and find that we can set the levels of detail (LOD) and separate the whole scene into four different levels. As user zoom in, GE will automatically change the level from one to four. Therefore, in this way our system automatically hide unimportant details or information that user doesn't want at that moment, e.g. when visualizing the whole network, the icons of individual genes are not visible.

The four levels in the current configuration are: species, network, pathway and entity. The visualization details are increased from level 1 to level 4 gradually and automatically when user zooms in. LOD enables user to derive novel biological insights for individual genes at entity level and for functional relationships at network level.

3.5. Advanced Interaction Methods

Although GE provides convenient navigation and edit abilities for the loaded network, they are not sufficient for an interactive network visualization tool since our proposed layered layout need fast ways to navigate each layer. Therefore we utilized GE's extended features to implement such advanced interactions in two ways. One is the Windows COM API; the other is the combination of network link and Update tag.

The Google Earth COM API allows third party applications to query information from and send commands to Google Earth. Through the IApplicationGE interface, applications can query the current viewport, control the 3D viewpoint, use KML features and determine the currently selected feature.

GE also officially supports the method of using <Update> element to change features in network link. In addition to pointing to files containing static data, a network link can point to data that is dynamically generated by a CGI script located on a network server. For detailed explanation and example of how it works, please refer to [20].

Both COM API and network link can provide the ability to create dynamic graphs, however, neither of them is perfect. Further development is needed to make both methods user friendly.

3.6. Integrated Control through GUI

As shown in the Fig 3.1 of the system framework, the user can control all the modules from MetNetGE's GUI. In order to provide user a clean view of control, the main GUI dialog is designed as small as possible, and it is shown as a floating dialog on the upper left corner of the screen. Each type of user actions is arranged to different tabs on the GUI window.

The available actions from GUI include loading the data files, customize the layout and color scheme, and set visibility of certain part of data. As described in section 3.6, user can select nodes that they are interested, and highlight their neighbors or nodes with high correlation in the 'Highlight' tab (See Fig. 4). The other feature of GUI we are currently developing is a general purpose list and search view of entities. The reason is that the default tree list view from GE will list every KML elements in the file, but what we want to see is the conceptual nodes and edges.

Chapter 4. Aligned 3D Tiered (A3T) layout of Pathway Visualization

Although Arena3D [20] and BioCichlid [21] have published algorithms of showing staggered layers in 3D, MetNetGE features an aligned 3D tiered layout. By separating the nodes into different layers according to their types, we can provide a clearer structure of metabolic pathway on the metabolite layer, or signaling pathway on the protein layer. This is the main reason why layers need to be used and aligned.

4.1. Layout Algorithm Description

The pathway diagram is denoted as a graph G, where G=<V, E>. V is the set of all nodes, and E is the set of all edges in the graph. We also separate the nodes into subset $V_i, i = 0,1,2,3$ where V_i represents all nodes on i^{th} layer. Each layer can contain any subsets of nodes. For our specific data of pathway networks, one of the most common choices is to divide nodes by type. Thus, V_0 represents all nodes on metabolite layer, V_1 represents nodes on polypeptide layer, and V_2, V_3 on RNA and DNA layer respectively. We further define the subgraph $G_i =< V_i, E_i >$, where $E_i =< edge(u,v),\ u,v \in V_i >$. Thus subgraph G_i is composed of all nodes from V_i and edges within V_i.

The layout algorithm in Arena3D is tier-independent because each layer lays out nodes independently. The links between layers are then drawn as lines between layers. The algorithm is described below:

```
TierIndependent(G):
  For ( i from 0 to 3):
    G_i = subgraph(G, V_i)
    Layout( G_i )
  End For
  Connect remaining edges in {G − ∪_{i=0to3} G_i}
```

One major difference between the MetNetGE layout and previous ones is that our node placement is based on one major plane, rather than computing each layout independently. We compute the layout of nodes on the subgraph of major plane first, and then set other nodes based on their relation to the nodes that have already been placed. The algorithm works like the following.

TierDependent(*G, imp*): //The algorithm for A3T layout. Imp: id of important plane.

$G_{imp} = subgraph(G, V_{imp})$

Layout(G_{imp})

NodeSet $T = V_{imp}$ // Represent nodes that are already positioned.

NodeSet $R = V - V_{imp}$ // Represent the remaining nodes that need to be positioned.

While (*R* is not empty):

Find every node $v \in R$, and *v* is connected by node $u \in T$, and *v, u* are **not** on the same plane, put all such *v* into NodeSet P_a

For every $v \in P_a$, place *v* with the same *x, y* position as *u*, so it's directly above or under *u*.

Remove P_a from *R*, and add P_a to *T*

Find every node $v \in R$, and *v* is connected by node $u \in T$, and *v, u* are on the same plane, put all such *v* into NodeSet P_b

Get the graph G_b which is composed by nodes of P_b, *u* and edges between them

Layout(G_b, with the positions of *u* fixed.)

End while

The A3T layout algorithm allows input to choose arbitrary plane as the important plane. However, in the context of pathway diagram, the pathway type already suggest good plane to use. Metabolic pathways will use metabolite plane as the important one since the metabolites consist the major structure of the pathway. Signaling pathways, on the other hand, will use polypeptide plane as the important one since the proteins are playing important roles in those pathways. Following sections will give example of these two types of pathways.

4.2. Complexity Analysis of 3D Tiered Layout algorithm

Assuming the input network has n nodes, the number of edge is $O(n^2)$. Assume the 3D Tiered layout will split the node set into k layers. Then each layer contain $m=n/k$ nodes. The complexity of layout algorithm G we use to generate layout for one layer is $G(n)$, and it can range from $O(n)$ to $O(n^3)$ depending on which algorithm we choose to use. For simplicity, assume we use force-based layout which is $O(n^3)$. Then, if we use this layout G on whole pathway, $G(n) = O(n^3)$.

In the 3D Tiered layout, the complexity is $T(n) = G(m)*k = O((n/k)^3*k = O(n^3/k^2) < O(n^3)$. In a typical data set, k is quite small, the gain in complexity reduction is not so significant. However, if this algorithm is used in dataset that has large value of k, the computational complexity can be significantly reduced. The usual range n,m,k of real pathway data are summarized as in the following Table 1:

Table 1 The typical value range for nodes, edges and layers in biological pathway.

Variable	*n* nodes	*t* edges	*m* nodes/layer	*k* layers
Range	50-300,	50-300	20-100	2-4

4.3. Layout Example of Metabolic Pathway

The utility of the A3T layout can be shown using one example of a typical metabolic pathway. For comparison, we first use Cytoscape's organic layout which is a good force-based layout to show the pathway "*ethylene biosynthesis and methionine cycle*" from MetNetDB[33]. The resulting drawing is shown in Fig 4.1left. The drawing appears to contain some pattern such as a closed cycle. However, those patterns are not easy to be detected.

Fig 4.1right shows the result of our A3T layout for the same pathway. It's clear that there is a cycle on the metabolite layer which means it is a metabolic pathway with feedback. The three blue edges coming from the polypeptide layer shows that those three protein complexes are catalyzing the three metabolic reactions above them. Other nodes on the polypeptide layer represent the proteins that compose those protein complexes, and each protein is translated and transcribed by the corresponding RNA and DNA respectively.

The advantage of the A3T layout in this example is clear: the major structure of this pathway become obvious. Moreover, if user navigated to the metabolite layer, they will find the drawing resembled the traditionally 2D layout, which is familiar to biologists.

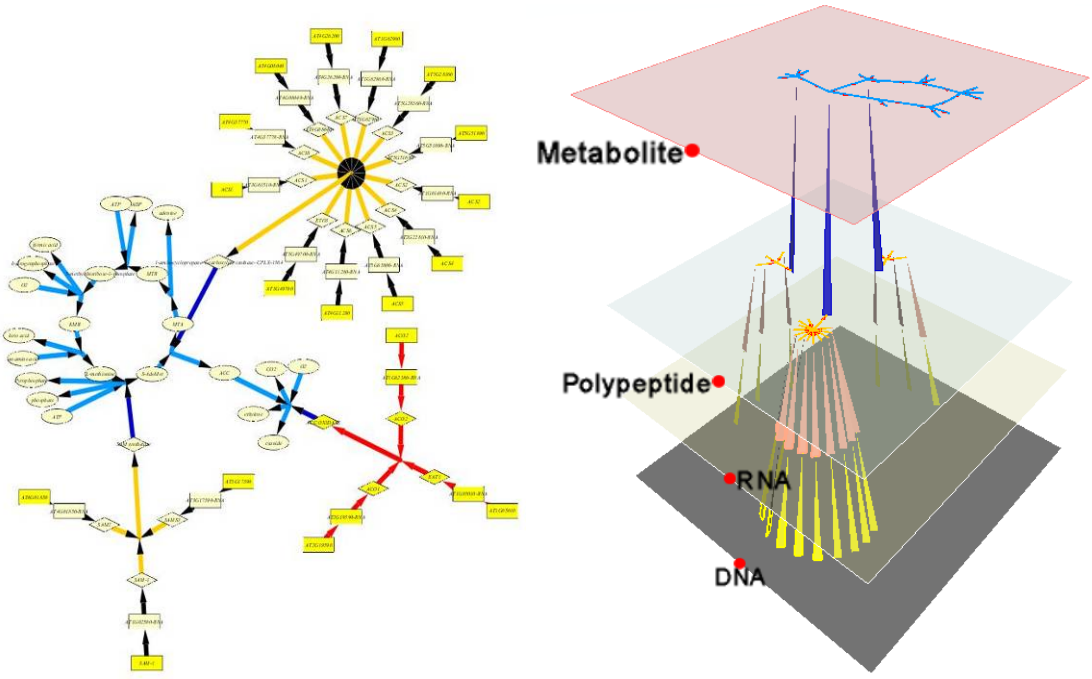


Fig 4.1 Visualize one metabolic pathway.

The pathway *ethylene biosynthesis and methionine cycle* is drawn using 'Organic' layout in Cytoscape (left) and 3D tiered layout in MetNetGE (right). The metabolite layer is chosen as major plane. It is clearly shown in 3D tiered layout that the pathway present circular structure in metabolite layer and three protein complexes catalyzed metabolic reactions (blue edge from protein layer to metabolite layer).

4.4. Layout Example of Signaling Pathway

An example of signaling pathway, the *ethylene signaling* pathway, is layered out based on polypeptide layer. This pathway is much more complex than the previous one. *Ethylene signaling* is one typical signaling pathway of Arabidopsis [3]. Fig 4.2 left is still the organic layout from Cytoscape. Red and green lines represent negative and positive regulations respectively. This view does not show any clear structure. The 3D tiered layout for this pathway is shown in Fig 4.2 right. Some interesting features immediately catch our eyes. For example, there is one metabolite (ethylene) negatively regulated many proteins. It is also noteworthy that one protein (erf1) positively regulated many RNAs. The pink and yellow lines represent translation and transcription links respectively.

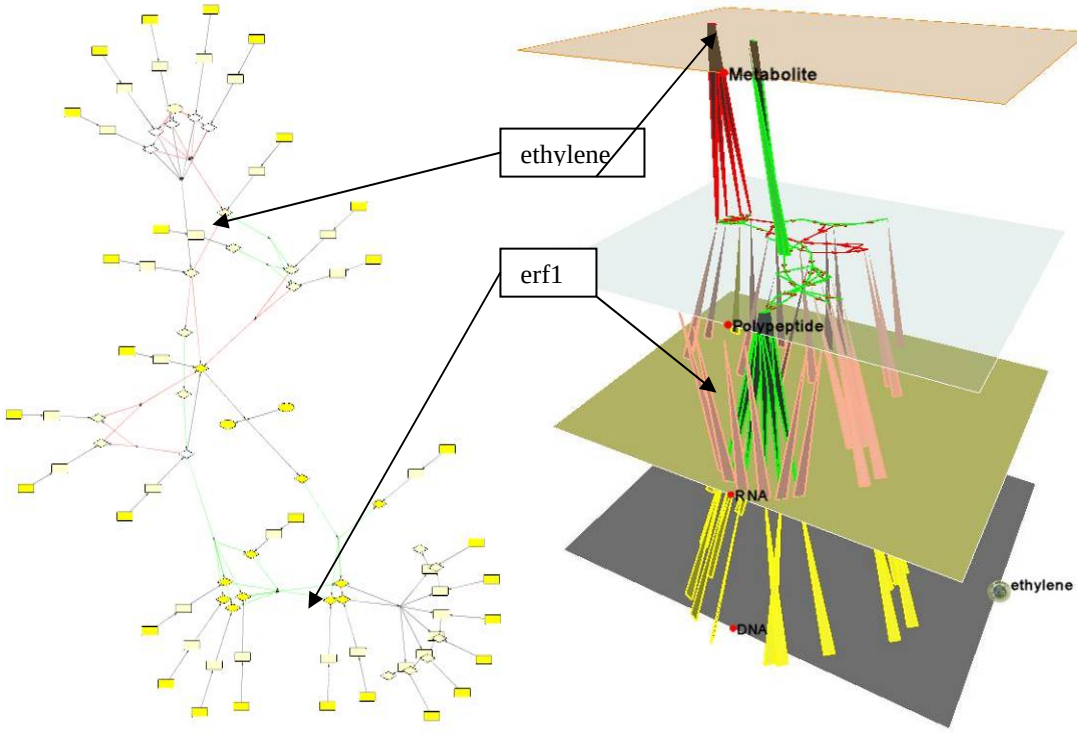


Fig 4.2 Visualize one signaling pathway.

The pathway ‘ethylene signaling’ viewed in ‘Organic’ layout in Cytoscape (left) and 3D tiered layout in MetNetGE (right). The 3D tiered layout revealed several interesting features which can not be easily seen from ‘Organic’ layout. For example, there are two metabolites (ethylene and ATP) that regulate many proteins, and one protein (erf1) activates many RNAs.

Chapter 5. Ontology Visualization using Enhanced RSF Technique

Our goal in MetNetGE is to follow Shneiderman's information-seeking mantra, "overviews first, zoom and filter, and details on demand". To give the user a meaningful global view, MetNetGE utilizes the pathway ontology to hierarchically organize the pathways. The ontology presents a directed acyclic graph, where many parents may point to the same child.

In the rest of this chapter, we will use graph terminology to describe the ontology and our visualization techniques. Thus the term "tree" means the data structure, but not the plant. Also, "leaf" node means the node in the tree structure that does not have any children, where "non-leaf" node means the node with at least one child, and is not related to the organism of a plant.

5.1. Visualize Tree Structure of Ontology

Among all the various tree visualization techniques, we implemented the radial space-filling (RSF) technique [26] because it effectively utilized the screen space and showed the hierarchy clearly. In addition, in RSF each non-leaf node has its own region, which provides the ability to map cumulative values onto those regions.

Researchers in economics have utilized 3D RSF to study hierarchical time-dependent data [34]. Their work inspires us to utilize the RSF to draw ontologies. To the best of our knowledge, there are no applications of 3D RSF drawing in biology field.

RSF visualization of a pure tree uses the following rules:

- Each circular region represents one node in the tree. The leaf nodes must be placed on the edge of the drawing and the root node is placed at the center.

- Each circular region has five variables: sweeping angle, depth, radius length, height, and color.
- The sweeping angle of a leaf node is determined by an attribute of the corresponding pathway. In our case, we have set each pathway to an equal weight, thus spanning the same angle.
- The sweeping angle of a non-leaf node is the sum of all its children's sweeping angles.

Initially, we use structure-based coloring [26] to convey more hierarchical sense, where the leaf node regions are colored according to the color wheel and the non-leaf node regions are colored as the weighted average of its children's color. We also set the height of each region proportional to the height of the sub-tree rooted at that node. Since color and height only affect the individual region, we later use them to map experimental values (see Chapter 6).

Fig 5.1 shows a typical tree with eight leaf nodes and five non-leaf nodes, labeled as graph G1. The bottom figure shows the result of using RSF in 3D on graph G1. Non-leaf nodes correspond to pathway categories, e.g. "A" may represent *acid resistance*. The leaf nodes represent the pathways, e.g. "A2" may represent *arginine dependent acid resistance pathway.* In this example, we use uniform radius length and structure based coloring, and map the height of the subtree to the region's height.

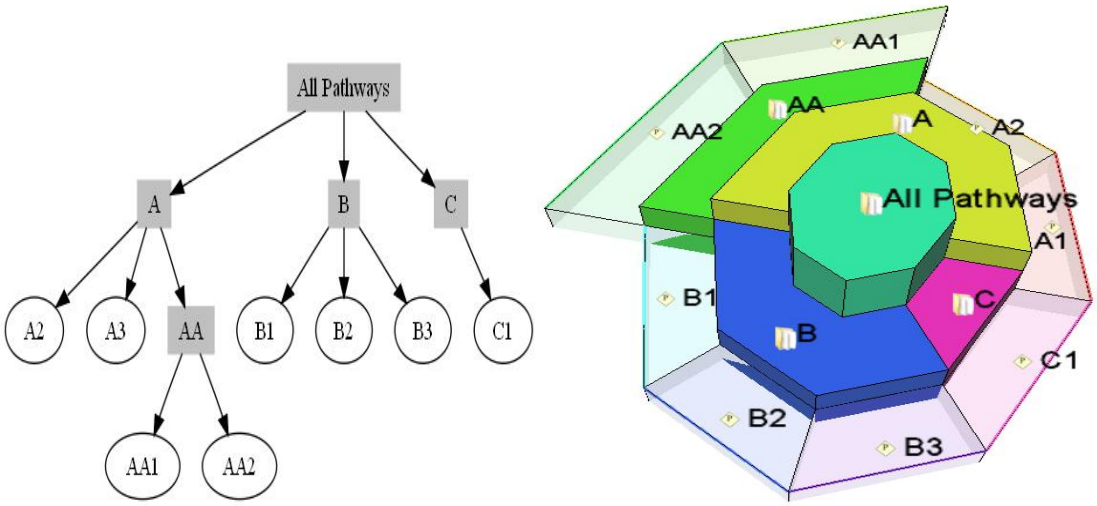


Fig 5.1 Visualization of graph G1 with tree structure.

Graph G1 of hypothetical relationships among leaf nodes (pathways) and non-leaf nodes (pathway categories), drawn in dot layout (left) and RSF (Radial Space-Filling) layout in MetNetGE (right).

5.2. Visualize Directed Acyclic Graph of Ontology

Now, consider the graph G2 (Fig 5.2), which adds four non-tree edges to G1. We use the metaphor of “satellite orbits” to represent these cross links. For each tree node which has at least two parents, one orbit is drawn on the layer of that node. We draw a blue edge, called the uplink, from the center of the node’s region to the orbit. We call the parent who connects the node in the spanning tree as the major parent and other parents as minor parents. The region of each node is placed under the region of its major parent. Then for every minor parent, we draw a green edge from the center of its region to the orbit of the child, and call this edge the ‘downlink’ (Fig 5.2). The connections between links and orbits are called access points.

To help viewers find and trace interesting cross links, the orbits need to be distinguishable from one another. We first restrict orbits to only span in the middle area of each layer, thus leave a visually apparent gap between orbits in adjacent layers.

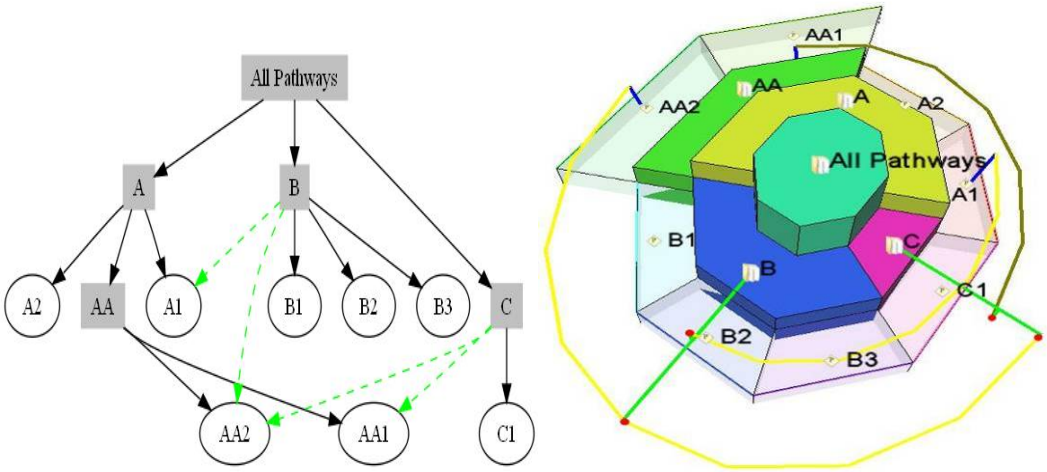


Fig 5.2 Visualization of graph G2 with non-tree structure.

Graph G2 drawn by dot layout (left) and ERSF (Enhanced Radial Space-Filling) layout in MetNetGE using structure-based coloring (right). In the dot layout, green dashed lines represent non-tree edges. In ERSF layout, yellow orbits and green, blue links represent non-tree relations. E.g. the green line extruded from C contains two red-dots: the inner one intersects with orbit of AA1 and the outer one intersects with orbit of AA2. The above orbits mean that C is the minor parent of both AA1 and AA2.

Then, to distinguish orbits in the same layer, our algorithm puts them at different heights and distances from the center. We sort the orbits by the number of downlinks. The orbit with most downlinks will be placed as the most distant and highest. This arrangement can help users answer questions like “Does the pentose phosphate pathway belong to many categories?”

Coloring strategies can help visually divide orbits that are located on the same layer. Biologists’ feedback suggested that using the child’s color or different hues can help distinguish orbits especially when the regions of categories are dimmed. In our pseudo example (Fig 5.3), the regions for non-leaf nodes are set as transparent while the orbits are set the same color with the child’s region. We call this mode orbits highlighting mode.

As we can see from the examples (Fig 5.2, Fig 5.3), visualizing the orbit metaphor with RSF has several advantages. First, this design clearly distinguishes between spanning tree relationships and non-tree edges. Second, compared to tree-maps with a crosslink overlay [29], there are much fewer edge-crossings. Third, all downlinks of a

parent share only one link edge. Thus, the total length of those edges is the same as the length of the longest link. This property reduces the graph complexity, especially when a child belongs to many parents or one parent is the minor parent for many other children nodes.

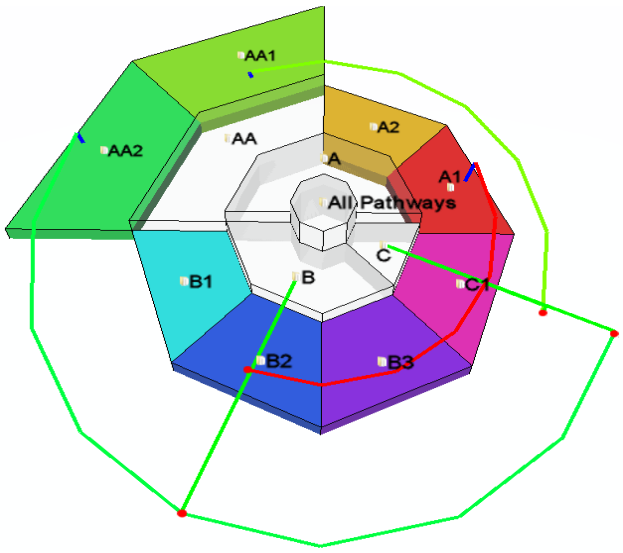


Fig 5.3 Highlight orbits of graph G2.

The regions for non-leaf nodes are set as transparent while the orbits are set the same color with the child's region, thus user can easily trace the orbit based on the color, e.g. the red orbit came from the red node A1.

Chapter 6. Mapping Gene Expression Data on Ontology

6.1. Map Average Expression Value and Coefficient of Variation

Biologists studying the large scale gene expression data sometimes want to know which pathways or categories are highly expressed in a certain condition. For example, they may ask questions like “Which pathway is highly expressed when we knock out a specific gene?”

Many tools provide a partial ability to answer the above questions. For example, Cytoscape allows users to map expression values to node color on the whole *E. coli* network, and the viewer can detect which parts of the network are highly activated. However, it takes much more effort to further understand what pathways are involved in such complex network.

Mapping the average gene expression value of a pathway onto its region’s color, enables the biologist to detect which pathways or categories are highly activated in a certain experiment.

The height of the region can be mapped to other values; e.g. the Coefficient of Variation (CoV) is also one interesting feature to consider. CoV describes how much each gene changes its expression value among many experimental conditions. The definition is following:

CoV= Standard Deviation of that gene / Mean of that gene

$=100 * \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2} \Big/ \mu$, where $\mu = \sum_{i=1}^{N} x_i \Big/ N$ and N is the number of conditions.

Since each pathway or category contains many genes, we provide options to show either the maximum CoV among these genes, or to show the average of CoV. We also

design a novel attempt to map two values simultaneously on the region's height by tilting the region. For example, the average CoV can be mapped to the height of the inner side of the region, while the max CoV can be mapped to the height of outer side. In this way, we can easily detect regions that tilt a lot which means some genes under those regions have very high CoV compared to other genes in the same region.

Fig 6.1 shows the result of mapping expression values on color and CoV on height for the pseudo data, and the result for real dataset is presented in the Result section. MetNetGE also use animation to show the values for a series of experiments, e.g. one time-series experiment with 7 time points will be presented as animation with 7 frames. User can use either the time controller from Google Earth or the animation control panel in MetNetGE to control the animation.

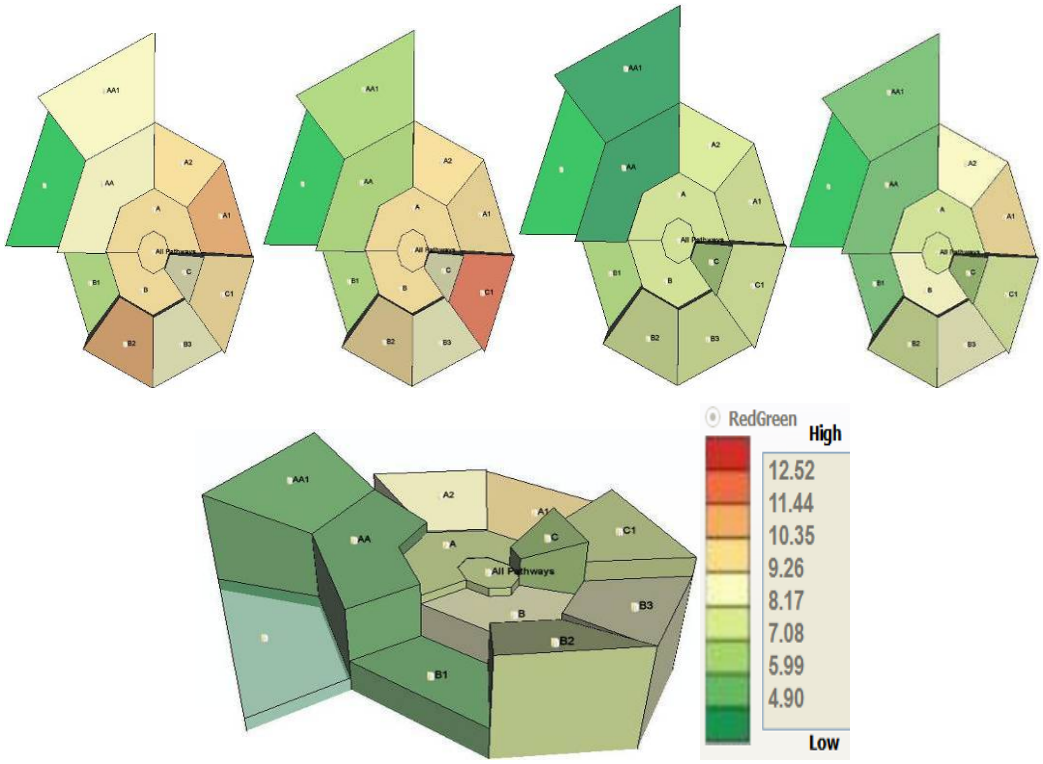


Fig 6.1 Mapping the gene expression data on pseudo dataset.

Pictures in first row show 4 frames of the animation where region color represents average gene expression value. Picture in bottom left shows the tilted view of this data, where high region shows high average CoV.

6.2. Map Differentially Expressed Genes on Ontology

The strategy of a biological scientist performing an omics study is typically to look for what parts of the network show significantly different measurements across different conditions. Questions like ‘Which pathways or categories are most changed under anaerobic stress?’ can be addressed by mapping the values onto the whole network.

Since biologists are more interested in the genes that are differentially expressed rather than the average express values, we can also map that information on the ontology drawing. We first define a threshold, e.g. 0.7 fold changes; then every gene that changes expression value greater than the threshold is considered differentially expressed. Then we count the number of up and down regulated genes for each category and pathway. To show the total number of differentially expressed genes, we map the log value of that number on region’s height. Then, we calculate the ratio of up/down regulated genes, and map it on the region’s color. The result section shows the view of this mapping.

6.3. Map Over-representation p-values on Ontology

In most of the experimental data analysis tasks, biologists are not simply interested in the expression value; instead, they are more concerned about some statistical results based on those raw data. For instance, one of our biologist collaborators has been analyzing the over-representation of pathway categories. One typical working scenario is that: first, she selected a group of genes which are highly expressed in one specific experiment condition, or are differentially expressed between two conditions. Then she used a statistical test, e.g. Fish Exact Test, to calculate p-value for every pathway and categories. After that, she viewed the category and p-value pair in the excel file, sorted and found the ones that looks interesting. Since the data in excel file didn’t contain the ontology, she had hard time to make some meaningful discoveries of the data. To help her

better understand the p-value and the categories, MetNetGE implemented all the required functions to visualize p-value on the ontology drawing, e.g. selecting genes within desired value range, performing statistical test, and mapping p-value on region colors. The visualization result is shown in the Result section.

Since there are many tools to help biologist in selecting interesting genes and performing varies statistical test with experimental data, we do not want to duplicate the functions of those tools. Therefore, MetNetGE can import external list of genes, or statistical test results in the simple CSV (Comma-Separated Values) format. We also provide simple python interface to let other developers to implement their statistical test method in python module and used in MetNetGE.

Chapter 7. Visualization Results

In this section, we present the pathway gene ontology for *E.coli* and 3D Tiered layout for Arabidopsis pathways that illustrate how MetNetGE can be helpful to gain insight in ontology structure and individual pathway. We will start our example with a typical usage scenario.

7.1. Pathway Ontology Visualization

We illustrate how can use ontology visualization module to explore the pathway ontology of *E.coli* from EcoCyc[11]. The EcoCyc ontology contains 442 nodes, where 289 of them are leaves. It also contains 508 edges, where 67 are non-tree links. The Graphviz [35] provide the 'twopi' layout which is considered very good at showing hierarchical structures. However, when used for this ontology, as shown in Fig 7.1, the hierarchical structure can hardly be seen because the non-tree edges distorted the structure.

Other popular ontology exploration tools, e.g. OBO-edit [13], can not represent this dataset too. For example, the TreeViewer in Obo-edit becomes a very long list of ontology names, and the Graph Editor becomes a short but extremely wide tree due to the high width/height ratio of the pathway ontology. In both editors, the global context of the ontology is missing and the non-tree edges are not obvious.

MetNetGE uses the ERSF layout to represent this ontology with structure-based coloring, as in Fig 7.2. Several interesting features of this data set stand out. First, the ontology has a very low height, i.e. the maximum distance from the root to the deepest leaf is only 6. Second, the ontology is not a tree structure, because the existence of orbits indicating that several nodes have multiple parents. The orbits are concentrated on the third layer, and

one category (*methylglyoxal detoxification*) contains many children in other categories. Furthermore, there is no child that belongs to more than four categories.

The names and other details of ontology terms can be viewed by simply zoom-in the view through Google Earth. It's clear that some categories contain much more pathways than others, e.g. Biosynthesis contains almost half of the pathways.

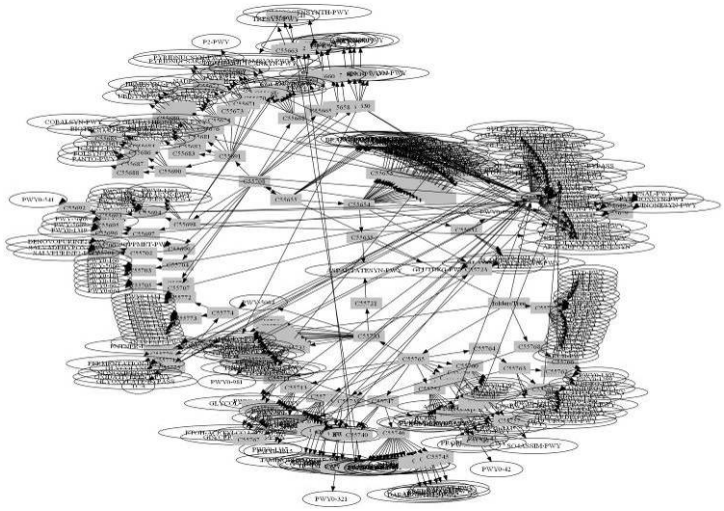


Fig 7.1 Pathway ontology from EcoCyc using the 'twopi' layout from the Graphviz software, the hierarchical structure can hardly be seen

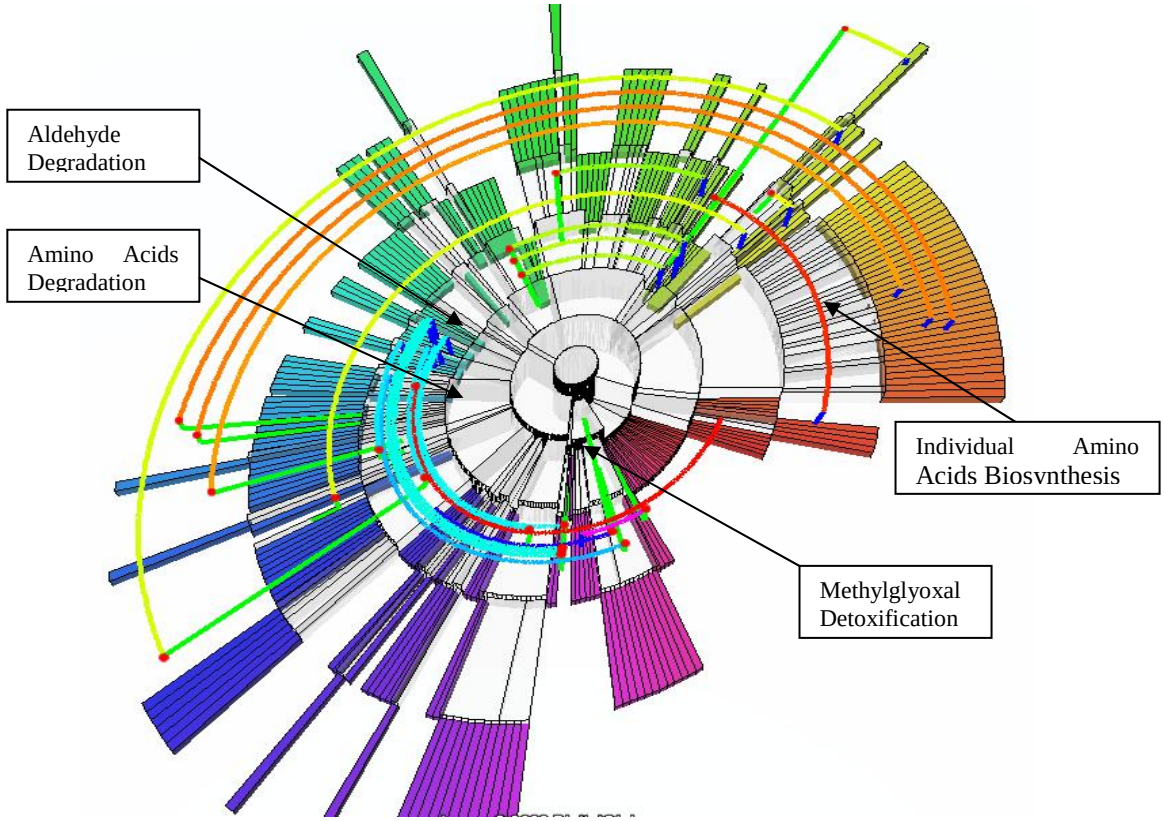


Fig 7.2 Pathway ontology shown with proposed ERSF layout.

It's clear that the ontology has hierarchical structure, and the height is 6. There are many pathways that belong to at least two categories, e.g. three pathways from *Individual Amino Acids Biosynthesis* (on right) also belong to the category *Amino Acids Degradation* (on left). Also many pathways from *Aldehyde Degradation* (on the left of 3rd layer) belong to category *Methylglyoxal Detoxification*. This kind of multiple inheritance information is hidden from most of other visualization methods.

MetNetGE also provides functionalities similar to OBO-edit, i.e., it also has a Windows Explorer™-like tree viewer and it is linked to the RSF drawing. Users can search the name for specific ontology node, as usually does with OBO-edit, and then locates the resulting node in the tree viewer. Furthermore, user can choose to show all the relations between the selected node and other nodes in the ontology. Fig 7.3 shows an example of above functions. It's clear that the selected category node ('*Amines and Polyamines Degradation*') shares children with many other category nodes, e.g. *'Sugar Derivatives Degradation'*. This kind of knowledge is also not easy to be discovered by other visualization methods.

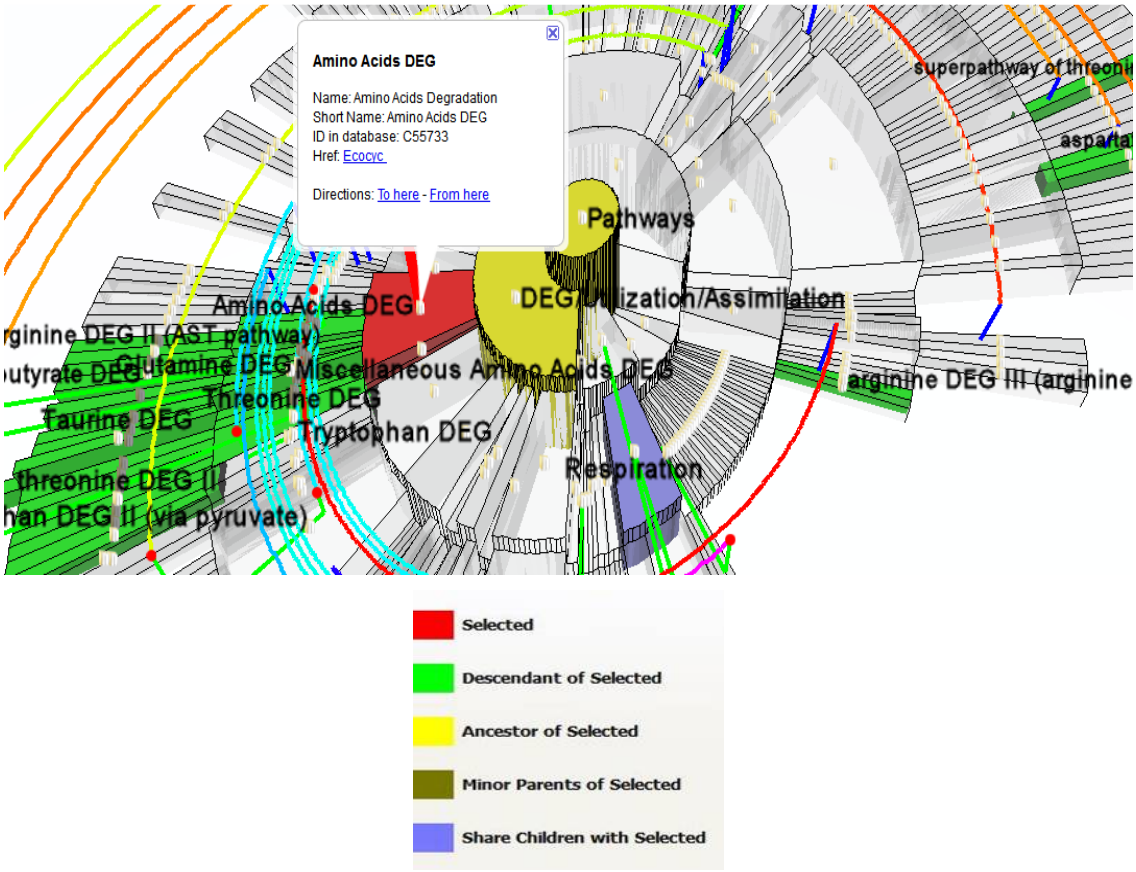


Fig 7.3 Related pathways/categories of a selected category.

Amines Acids Degradation is selected (in red), its descendents (in green), ancestors (in yellow) and other categories that share child with it (in blue) are shown.

Simple interactions like rotation, pan is really helpful when tracing the ancestors or decedents of a selected category. The related ontology terms can be easily read from the drawing. In other tools, users need to perform several scrolls and expand actions in the explorer list to find all decedents.

7.2. Mapping Omics Data on Pathway Ontology

After viewing the structure of pathway ontology, Tom realized that he can actually map the experimental data on the ontology. He selected one experimental data, which compares gene expression profiles of *E. coli* grown with or without *Acacia mearnsii* (black wattle) extract under anaerobic condition to study tannin resistance strategy [36]. The data contains two replicates under two experimental conditions, and compares gene expression of 4217 genes.

He normally performs this task using Excel. In Excel, Tom inputs the ontology terms on each row on first column. Then he calculates and fills in the average gene expression value on the second column. Then he fills other information on other columns. If he wants to see which pathways or categories have the highest expression value, he can simply sort the data by the second column, then the categories he want to know will become top rows. However, in this spreadsheet view, Tom lose the relationship between those categories, e.g. if pathway P1 and P2 are both highly expressed, he wants to know whether they belong to the same category, sadly, it's hard to tell it in Excel. What he can do is to search P1 and P2 in EcoCyc website, and see their relation.

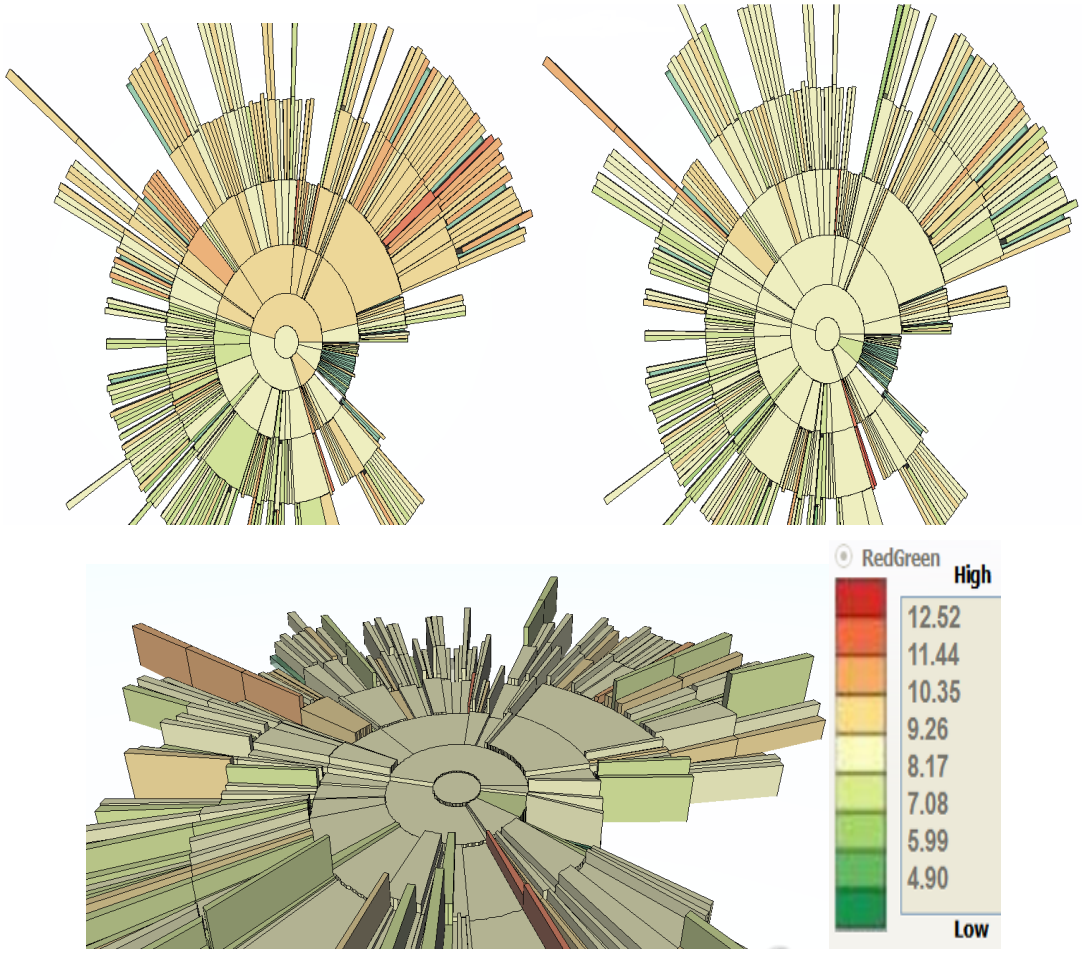


Fig 7.4 Average expression values are shown for each condition.

The orange and red color in condition 1 represent that many categories have much higher expression value in condition 1 (left) than in condition 2 (right). When the view is tilted (bottom), the categories with high Coefficient of variation (CoV) is shown by their higher height.

In MetNetGE, Tom maps the average expression value to the color of ontology regions, and maps the coefficient of variation to the height of those regions. After loading the data, MetNetGE generate animation frame for each replicate. Then he can use GE's animation slider to identify highly expressed categories or pathways, and those with high CoV for each replicate. Fig 7.4 shows some frames of the animation. Reddish regions represent categories with high average expression value, and greenish regions show the ones with low average expression value.

When playing the animation frame by frame, Tom finds out that in condition 1, most of the categories and pathways have high average expression value, especially the ones on upper side of the ontology. However in condition 2, many categories' average expression value become low, it means a lot of genes in many categories are expressed more in tannin treated condition (condition 1). He can easily verify this phenomenon by tilting the 3D view and see that many regions are high meaning that genes in them varies a lot during this experiment.

He then wants to confirm the discovery that most of the categories are down-regulated. So he maps the differential expression directly on regions color, and maps the total number of differentially expressed genes on height. Since in many regions, some genes may up-regulated (the expression value increased during the experiment), but some other genes may down-regulated. As a result, Tom maps the ratio of up/down regulated genes on color. The visualization result is shown in Fig 7.5. He can see that most of the upper side of ontology categories is down-regulated, while some categories on the lower-left side are up-regulated.

As he zoom in, we can see the details of those categories, e.g. Amino acids Biosynthesis has totally 62 genes that are differentially expressed, while 58 of them are

down regulated. When viewing the super pathway category, two super pathways immediately catch his eyes because they have many more genes that are differentially expressed than others (Fig 7.6). They are *superpathway of histidine, purine, and pyrimidine biosynthesis*, and *superpathway of chorismate*. Among the few categories that are up-regulated, he can see *Sugar Acids Degradation* has all 8 genes up regulated. Details about a specific category/pathway can be seen in MetNetGE by simply select it.

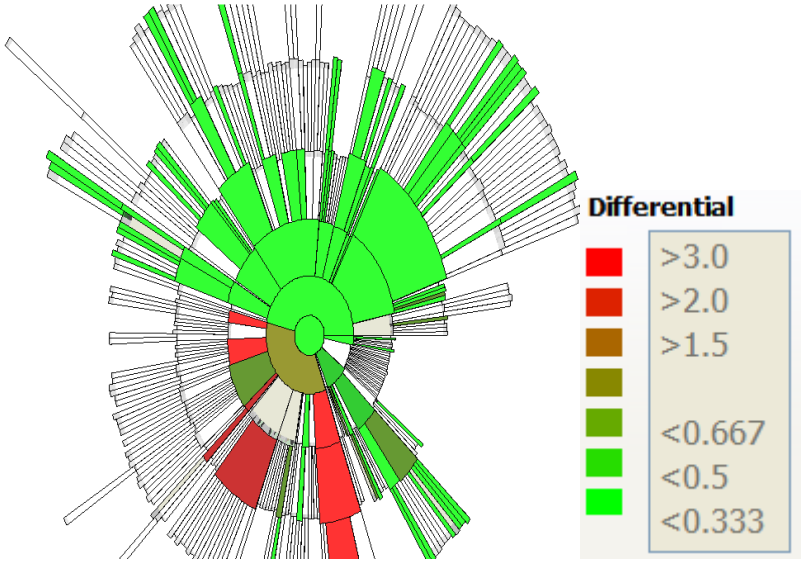


Fig 7.5 Differentially expressed genes mapped on ontology drawing.

Color indicates the ratio of up/down regulated genes; height shows the log value of total number of differentially expressed genes. We can see that most of the upper side of ontology categories is down-regulated, while some categories on the lower-left side are up-regulated.

Tom can also see the raw experiment values with the parallel coordinate plot and the spreadsheet for further investigation (Fig 7.7). From our view of ERSF layout, all the differentially expressed categories and pathway can easily be detected with the color indicating whether it is mainly up-regulated or down-regulated. Only a couple of those categories are listed in the paper [36] for this experiment, which means the proposed visual method can help us find more interesting features during an experiment.

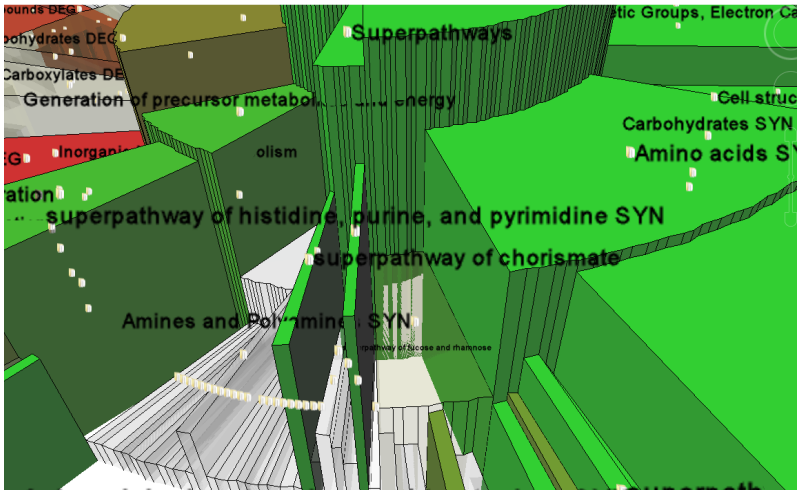


Fig 7.6 Zoom-in view shows that two superpathways (*histidine, purine, and pyrimidine biosynthesis*, and *chorismate*) have much more genes differentially expressed than other superpathways.

The other way to view the up or down regulated genes is to use over-representation. He can use any statistical method to select genes, and then calculate p-values for over-representation among pathways using Fisher Exact Test. The calculated p-values can be loaded on the ontology drawing in MetNetGE.

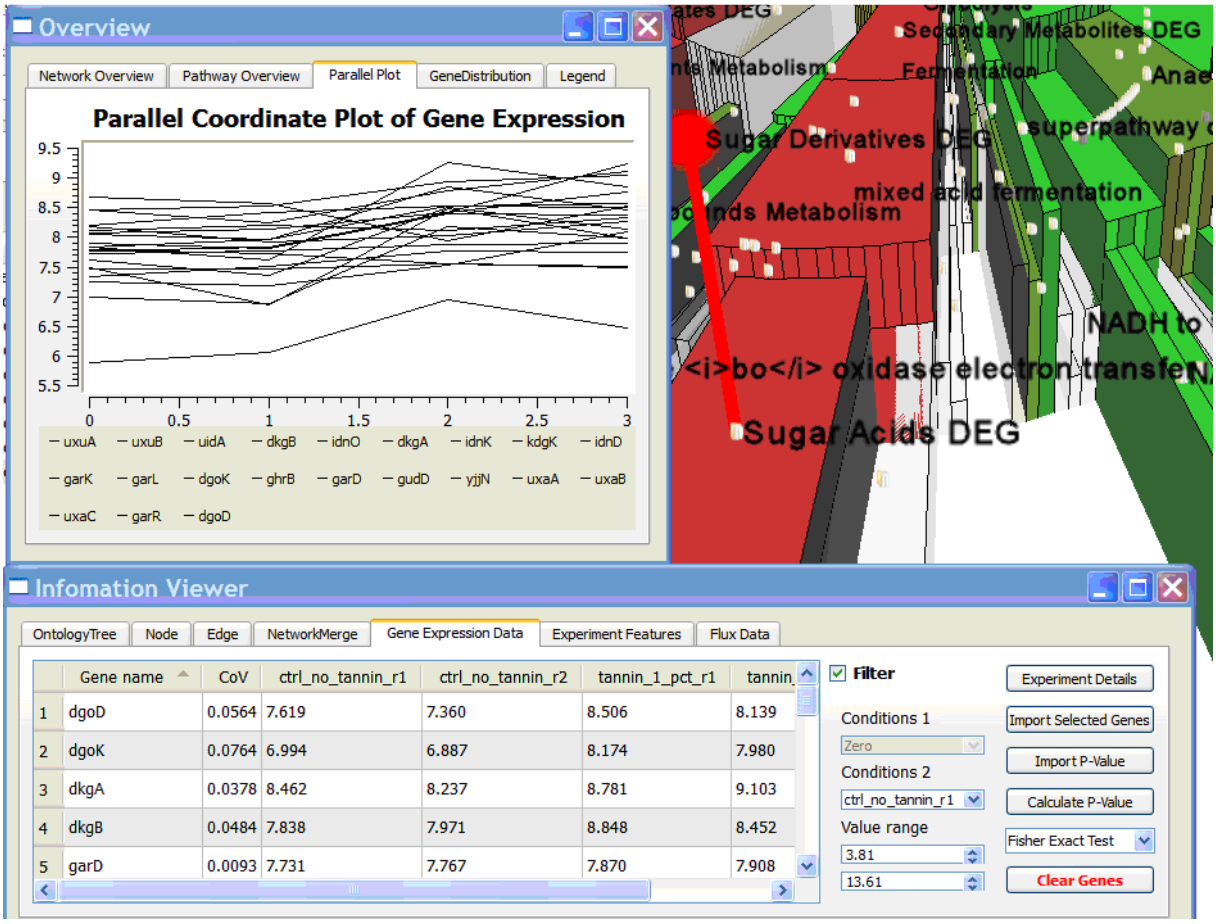


Fig 7.7 Parallel coordinate plot in MetNetGE.

After identifying the interesting category ‘sugar acids degradation’, user can add genes in this category, and view their values in both traditional parallel coordinate plot and the spreadsheet.

Chapter 8. A user study of visualizing ontology and experimental data in system biology

A paper submitted to the International Journal of Human Computer Studies

Ming Jia[1], Stephen Gilbert[2], Eve Syrkin Wurtele[3], Julie A. Dickerson[1,2]

[1] Dept. of Electrical and Computer Engineering, [2]Human Computer Interaction Program

[3] Dept. of Genetics, Development and Cell Biology, Iowa State University, Ames, IA, USA.

jiaming@iastate.edu, gilbert@iastate.edu, mash@iastate.edu, julied@iastate.edu

Abstract. The increasing volume of experimental data in biological research has posed several new requirements for the data visualization. Biologists need the visualization to map the whole experimental data onto ontologies to understand the effect on system scale. One proposed layout algorithm, enhanced radial space-filling (ERSF), was designed to meet these new requirements. To demonstrate that ERSF is more efficient than current tools regarding these requirements, we conducted a user study involving twenty participants. The study suggested that although ERSF requires longer learning times, it largely outperforms the compared tool in completion time in representative tasks. This is mainly attributable to the orbit-metaphor introduced in the ERSF drawing, which distinguishes normal edges and non-tree edges, and the efficient use of screen space to show experimental data.

1 Introduction

Linking large-volume experimental data with hierarchical ontologies that relate biological concepts is a key step for understanding complex biological systems. Biologists need an overview of broader functional categories and their performance under different experimental conditions to ask questions such as whether degradation pathways have many highly expressed genes, or which biological process categories are overrepresented in the data. These needs pose many unique requirements on the visualization of biological ontologies, such as being able to visualize an overview of an

ontology mapped with experimental data and clearly show the non-tree connections in ontology.

Current tools that visualize biological ontologies normally employ the traditional Windows™ Explorer-like indented hierarchical list, as are found in EcoCyc [11] and AmiGO [30], or node-link based layouts (see Fig. 1), e.g., OBOEdit [13] and BinGO [31]. These kinds of layouts are well suited for tens of nodes, but quickly become cluttered if hundreds of nodes are shown simultaneously.

To address these problems, the authors proposed the enhanced radial space-filling (ERSF) algorithm [37] that uses an intuitive orbit metaphor to explicitly visualize non-tree edges, and makes them appear differently than the major hierarchic structure. The ERSF, as well as other proposed algorithms, were implemented in a software package called MetNetGE [38].

A preliminary user test with the ERSF algorithm indicated that users preferred the ERSF solution to the traditional indented list and node-link based layout [37]. In this paper, we report the procedure and results of a larger user study comparing the ERSF and MetNetGE with a widely known software tool. The key finding of our user study is that, although ERSF requires much longer learning time, it largely outperforms the competing tool in our selected tasks in terms of completion time. Ontology Data and Visualization Requirements

An ontology is a formal explicit description of concepts, or classes in a domain of discourse [39]. Biologists use ontologies to organize biological concepts. The Pathway Ontology (PO) [9] is a controlled vocabulary for biological pathways and their functions. The PO is hierarchical, but it is not a pure tree structure because several pathways may have multiple parents. Ontologies are directed acyclic graphs and contain both tree and

non-tree edges. The non-tree edges are of particular interest since they represent pathways that perform multiple functions.

For example, the *E. coli* Pathway Ontology [11] contains 442 nodes, where 289 of them are pathways or leaves. It also contains 508 edges, where 67 (13.2%) are non-tree edges. Another feature typical of a PO is that the depth of the hierarchy is normally low, e.g., 6 for *E. coli*, which results in a very large width/height ratio (289/6=48.1).

In their daily research, biologists need to make sense of system-wide experimental data and wish to understand how the experimental conditions affect the underlying biology. One typical type of experimental data is transcriptomics (often referred to as gene expression data), which describes the abundance of gene transcripts during an experiment. The original data is typically a data matrix in which each row describes a gene, and each column records the expression level of genes under a certain condition, e.g., drought stress or a mutation.

Based on the data and tasks biologists perform, the basic requirements for the visualization of a Pathway Ontology and experimental data are:

R1. View the whole ontology on a single screen to gain global knowledge and the main hierarchical structure.

R2. View ontology details by navigation and/or interaction (zoom, pan, rotation).

R3. Map experimental data on the ontology so that they are easily visible and distinguishable.

R4. Clearly show non-tree connections.

1.1 Related Works in Visualizing Ontology Data

Biologists normally view ontology structure as an indented list, e.g., EcoCyc [11] and AmiGO [30]. One implementation of an indented list (Class Browser) is evaluated in [40] with three other methods (Zoomable interface, Focus + Context, and Node-link/tree). The indented list lacks the ability to show non-tree edges. Users presented with an indented list naturally think the underlying data is a pure tree structure.

Node-link based layouts are also widely supported. For example, OBO-Edit [13] combines an indented tree browser (Tree Editor) and a graphical tree drawing (Graph Editor) (Fig. 1) which uses the node-link based layout from GraphViz [35]. BinGO[31], a Cytoscape plug-in for analyzing Gene Ontology, uses the default 2D hierarchic layout from Cytoscape. The node-link based layout is very good at showing simple hierarchical structures (e.g. containing less than 50 nodes). However, when the number of entities increases, those layouts become cluttered and incomprehensible.

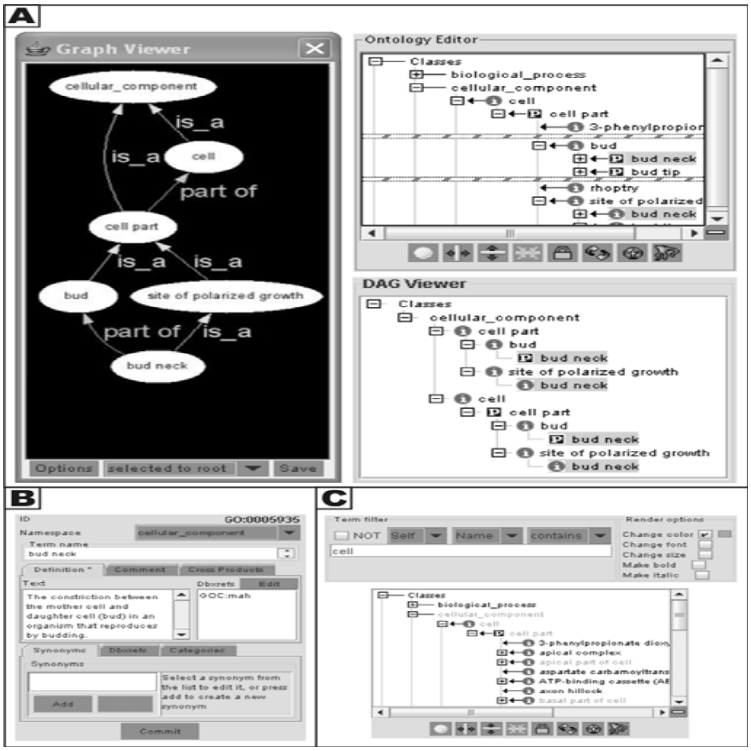


Fig. 1 OBO-Edit combines the node-link drawing (left) and indented tree browser (right) to represent the Gene Ontology.

Treemap based systems [41] are able to visualize the whole ontology with mapped data in one screen, and are suitable for identifying regions of interest. However, the hierarchical structure is hard to see in a treemap since it is a nesting-based layout which superimposes the child nodes onto their parent nodes [34]. Another limitation of the treemap is that it lacks a meaningful representation of non-tree edges, as indicated in requirement R4. As observed in [39], treemaps and other space-filling layouts normally duplicate nodes which have multiple parents. If the node being duplicated is a non-leaf node, the whole substructure rooted at this node will be duplicated as well. Therefore duplicating nodes in a hierarchic dataset may greatly increase a graph's visual complexity.

Katifori et al. [39] have also presented many tools and layout algorithms to visualize ontologies and graphs in general. For example, a hyperbolic tree [28] can handle

thousands of nodes. However, in a hyperbolic tree visualization, it is difficult to distinguish between tree and non-tree edges among hundreds of edges since they are all represented as links. Another disadvantage is that hyperbolic trees are not space efficient, and normally only a couple of pixels are used for each node. Therefore attributes (like gene expression data) mapped on nodes become hard to distinguish and interpret.

Space-filling methods are considered very space-efficient and are good for mapping attributes on node regions. Despite the disadvantages of rectangular space-filling (such as treemap), evaluations [42] find that radial space-filling (RSF) methods [26] are quite effective at preserving hierarchical relations.

The enhanced radial space-filling (ERSF) algorithm first extracted one spanning tree from the ontology data, and visualized it by the RSF method[26]. To represent non-tree edges in the ontology, ERSF algorithm draws orbits from the nodes which have multiple parents. This method makes the non-tree edges clearly stand out.

1.2 User study goals

The initial reason to implement our proposed layout algorithms is that we find the existing software tools are not good at enabling tasks on the ontology dataset in viewing and analyzing the whole topological structure and understanding the experimental data. We also closely observed how our biologist collaborators performed those tasks in their daily work, and understand the huge amount of manual work involved in using the existing tools. To test whether ERSF is effective in these tasks, we designed and conducted a user study with 20 participants (students in biology and computer science). The hypothesis is that the ERSF methods can help biologists to understand the relationships between changes in pathways and perform these analysis tasks more easily in terms of completion time.

We chose not to mimic every aspect and features of existing tools; doing so would be a waste of research time. Instead, we focused on improving usability and performance on the analyses that existing tools can't handle very well. As a result, the goal of our user study is to selectively pick the tasks in which users have trouble with existing tools, and see if their performance improves when using MetNetGE.

Selecting the proper tasks to test in our user study is also a hard problem on its own. One reason is that biological tasks are complex and time-consuming in general, and we don't want to require participants to spend too much time in the study (we want to limit the total participant time within one hour), or they will get bored and frustrated, which makes the evaluation less accurate. The other reason is that the core contributions of our algorithms are not confined to biological area. They can also be applied to general ontology visualization as well. Thus we can attract a broader range of participants if we keep the biological concepts to a minimum. The third challenge is that we want to use tasks that are actually useful and needed for our users.

Based on the above reasons, we focused our study tasks on pathway ontology and the omics data mapped on it, which can be easily explained to both biology and computer science students. We compared MetNetGE with the ERSF with a highly used existing tool, Cytoscape, a 2D graph display program. .

The details of the user study are described in Section 2. The result and analysis of the study are in Section 3. Section 4 discusses and evaluates the user study methodology and results, and proposes further improvements. Finally, Section 5 concludes the analysis.

2 Method

Our study included three steps. Participants were first given the tutorials of using both tools to visualize ontology data. Then, they used both tools to go through several tasks. Finally, participants completed an online post-study questionnaire.

2.1 Participants

We sent out recruiting emails (which includes the Study Consent Form) to the students in our research group, students in the biology department and students in the computer science department. There were 23 replies to the email, and 20 of whom actually participated in the study. All participants were graduate students and their ages ranged from 23 to 35. We gave $10 to each participant. Among the 20 participants, 7 (35%) were from the biology department and 13 (65%) were from computer science. There are also 4 female participants (20%) and 16 (80%) male participants.

2.2 Study design

The independent variable (IV) of this study was the software package used. One level was MetNetGE, and the other level was the compared software, Cytoscape. The dependant variables (DVs) were objective measurements of the participant's performance in completing the tasks, including completion time and the number of errors. Each participant used both tools, thus this was a within subjects design.

2.3 Terminology

In order to better describe the tasks, we list some important terminology below.

Pathway Ontology: The Pathway Ontology is a controlled term for pathways. It has a hierarchical structure, which consists of a tree structure and many non-tree edges. We will use graph terminology to describe the ontology. Thus the term "tree" means the data structure, "leaf" node means a node in the tree structure that does not have any children,

and “non-leaf” node means a node with at least one child. In our ontology, leaf nodes are Pathways, non-leaf nodes are Categories.

Descendant, Children and Parent: For a given category, children are nodes directly connected and under this given category. This given category is called the parent of these children nodes. Descendants are all nodes under a given category, including all indirectly connected ones. If a pathway is a descendant of a category, we say this category contains this pathway, and this pathway belongs to this category.

Highly related categories: If two categories have at least 3 common children, we call them a pair of highly related categories.

Level: The root of the ontology is on level 0. Every node directly under root is on level one. The child’s level is one plus its parent’s level.

Depth of the ontology: The depth is defined as the maximum level of leaf nodes.

The above terminologies used in the user study were first described in the tutorial section. In our pilot study, we found that users tend to forget terminologies and concepts. Therefore, those concepts were also presented in the hint section of each task.

2.4 Pilot user study

We conducted a pilot user study with two people in our institute who were both computer science students and familiar with the concepts of biological ontologies. Since we wanted to test many aspects of the tools, the original pilot study consisted of five parts. The first three parts analyzed the topological structure of a given ontology. To let the user get familiar with the tools and the ontology gradually, we started with a very simple ontology that contained only 13 nodes. The second part used a medium ontology with one hundred nodes, which is actually the Gene Ontology Slim, or GO Slim [43]. The 3rd part used a much larger ontology that contained about 500 nodes, and it was the

pathway ontology of *E.Coli* or *Arabidopsis*. In the 4th part participants examined the over-expression statistical significance (p-value) mapped on the medium ontology. The 5th part let participants work on the mapping of omics data on the large 500 node ontology.

However, the first participant in the pilot study spent a very long time in the tutorial part, and used up the one-hour time without even completing with one tool. As a result, we reduced most of the tasks for the second pilot study participant so she could finish the study within one hour. The detailed tasks will be listed in the sub-section Tasks.

During the pilot study, we found that the effort to analyze the topological structure of a large ontology using Cytoscape was so demanding that the participants got very frustrated and declined to work on it anymore. Although they could perform the tasks on this network fairly well using MetNetGE, we could not get a valid result for Cytoscape. For this reason, we removed the analysis of large networks task even though we believe this is where MetNetGE largely outperforms Cytoscape.

Cytoscape provides a wide range of 2D layout algorithms. To focus the study, we selected the best layout algorithm in Cytoscape for each task and always used that layout in the user study. Other user studies, e.g., [23], indicated that the force-directed Organic layout appears to be the best automatic layout for social network groups with around 50 nodes. However, when applied to the ontology in our user study, the Organic layout generated a graph structure as in Fig. 2, which was hard to interpret for our tasks. One major reason is that the existence of multiple inheritance edges distorted the layout, so the whole hierarchy can not been seen. Therefore, we chose the Hierarchic layout, which is widely used when analyzing hierarchical structure.

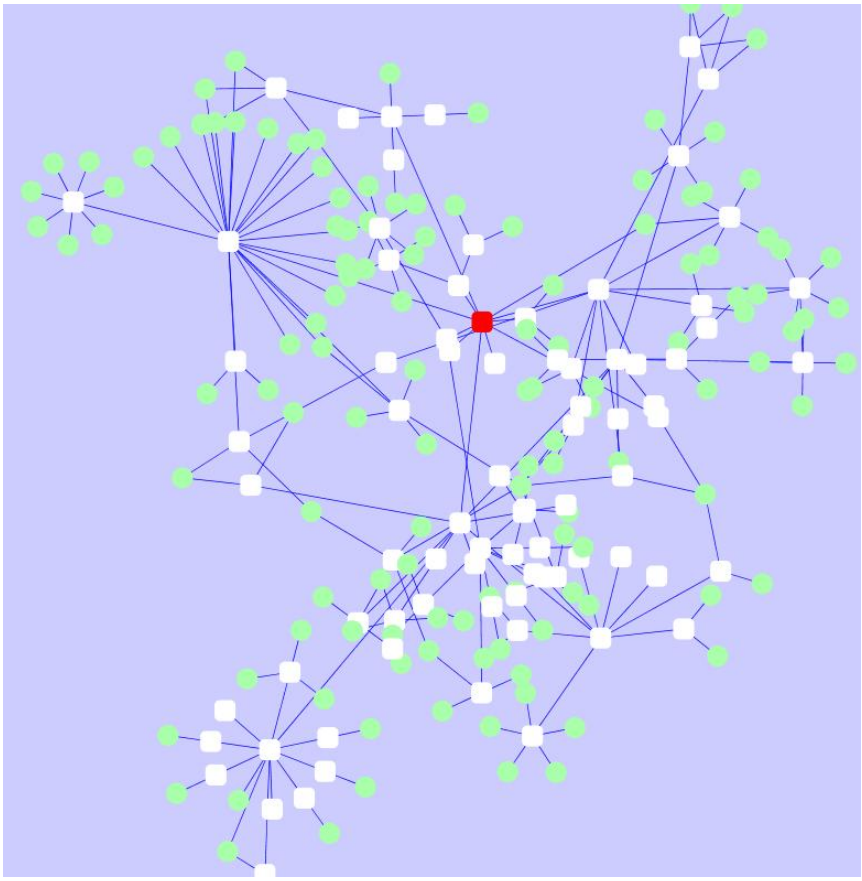


Fig. 2 The medium size ontology is shown in Cytoscape with Organic layout. The red rectangle is the root, the white rectangles are categories, and the green circles are leaf nodes. Due to the existence of multiple inheritances, the ontology structure is distorted, which makes it extremely hard to understand the topological structures.

2.5 Tasks

The final tasks used in the user study contained two parts. Part one concentrated on analyzing the topological structure of one medium sized pathway ontology (about 200 nodes). The ontology was extracted from the whole pathway ontology and modified to have some prominent features, e.g., added pairs of related categories. To prevent participants from carrying knowledge and the answer from the first tool, we used two slightly different datasets in each tool. Cytoscape used a pathway ontology from *Arabidopsis*, while MetNetGE used pathway ontology from *E.Coli*. We also modified the dataset so that the network used in MetNetGE was slightly more complex than the corresponding network in Cytoscape. For example, the medium ontology in MetNetGE

contained 218 nodes and 238 edges, while the one in Cytoscape contained 206 nodes and 226 edges respectively. The structure of the ontologies and the number of non-tree edges were similar.

Since a participant might have forgotten the concepts and visual cues and controls learned in the tutorial parts, we provided hints in both tools. Table 1 lists the tasks for part one and the hints for using MetNetGE.

Table 1 Part one of the user study tasks and hints for using MetNetGE.

ID	Task description	Hint in MetNetGE
1	Which category in level one contains the most leaf nodes in its descendants?	Find the category that has largest angle. Do not need to be the exact answer.
2	Which category in level two contains the most leaf nodes in its descendants?	Find the category that has largest angle in level 2.
3	What is the maximum depth of this pathway ontology?	The root has depth 0
4	Please find out one pathway (leaf nodes), which has at least 2 parents.	Try to find leaf node that has white links.
5	Please find 3 pathways, which each have at least 3 parents.	Looking for orbit that intersects with at least two blue links.
6	Can you find a pathway which has at least 6 parents?	Looking for orbit which has 6 red dots.
7	Do you observe any pair of categories that are tightly related (share at least 3 children)?	Try to find two pairs like this.

The first two tasks let participants find the categories in a certain level which had the most leaf nodes. This information is important for biologist to get an initial understanding of a given ontology. When a category has more leaf nodes it means that category is more complex, or we have more knowledge about its functionality. Tasks 4, 5, and 6 required participants to find the pathways which had multiple parents. Those pathways are important in the biology field, because those pathways may have several features. In other ontologies, e.g., a computer program class hierarchy, the classes which extend from multiple classes are also important. The related categories in task 7 shared at least 3

children. Those categories may have similar functionalities, and may present similar behavior in omics data. In a computer science class diagram, if two classes are related, software developers may consider re-factoring the codes of these classes, e.g., merging these two classes together or splitting them even further apart to reduce the duplication of their functionalities.

Part 2 of the tasks (listed in Table 2) focused on the analysis of omics data mapped on the whole pathway ontology. Normally, biologists would like to see the ontology nodes that have extreme values in one condition (e.g., much higher than other nodes). More often, they need to see which nodes changed values a lot across two different conditions. We cover both of these cases in our study. Besides being interested in the experimental value of individual nodes, biologists want to find if pathways in the same category have a similar value, which takes the ontology structure into consideration. As a result, we defined a region of nodes that consisted of one category and at least 3 of its children with similar values in certain conditions. We wanted participants to find such regions in a single condition or across conditions.

Table 2 Part two of the user study tasks and a hint for using MetNetGE.

ID	**Task description**	**Hint for using MetNetGE**
1	In condition 1, please find two regions which have very high value (pure red).	You can zoom out to see the overview of the ontology
2	Please find two regions which have very low value (dark green) in Condition 1, but have very high value (pure red) in condition 2.	
3	Please find two regions which have very high value in Condition 1, but have very low value in condition 3.	

One of the reasons to use the above tasks is that we wanted users to focus on the behavior of a group of nodes in a global scale. We hoped users could always zoom-out to see the whole picture of what's happening during the experiments, and what's changed.

These are the questions biologists always try to answer when analyzing real experimental data.

2.6 User study setting

Since neither software tool used in our user study demand significant computing power to run, we used a Lenovo™ T61p laptop with dedicated graphics processing unit (GPU) and 15-inch screen. We connected the laptop to a 24-inch LCD screen with 1920x1200 resolution, so the participants could view the ontology in larger screen and higher resolution. Since the names of ontology nodes are very long, writing them down on papers would have cost a lot of time, which would have made the measurement of task performance less accurate (e.g., the case where finding a pathway requires 10 seconds, but writing it down requires 20 seconds.). To solve this problem, we created a questionnaire for the task using Google Forms which let user copy and paste the answer into a web browser viewable on the laptop's screen. Participants could easily view the ontology structure on the large screen while filling out answers on the smaller laptop screen without switching applications.

Due to the difficulty of the tasks, participants sometimes needed immediate help if they forgot some concepts, or grew frustrated. Thus the observer always sat beside the participants and gave help when needed. For example, if the participant forgets what the color coding means in ERSF or Cytoscape, observer can assist them. The number of required helping moments was also recorded. Users were not asked to think aloud when performing the tasks because we didn't want to affect their performance.

2.7 Procedure

2.7.1 Learn tutorial network with both tools.

Every participant was first given the written consent form, and the form was signed before starting the user study. The participant then started the tutorial from online

instructions regarding a small ontology sample. In this section, participants read through the tutorials of each tool, and walk through some sample tasks. Since the concept and visualization metaphor of MetNetGE are non-traditional to even computer sciences students, we always started the tutorial with the Cytoscape version. The small ontology is shown in Fig. 3. The visualization metaphors are explained in the caption of the figure. Participants were given the same tasks as in part 1 of the real task, although the answers in this part were already given. Participants were encouraged to think about why the answers were correct for each task. We observed that many of the participants were confused about the concept of children and descendants. Thus the observers gave hints about those concepts when participants came out with the wrong answers. We recorded the time participants spent going through the tutorial section, and used it as an indication of learning curve.

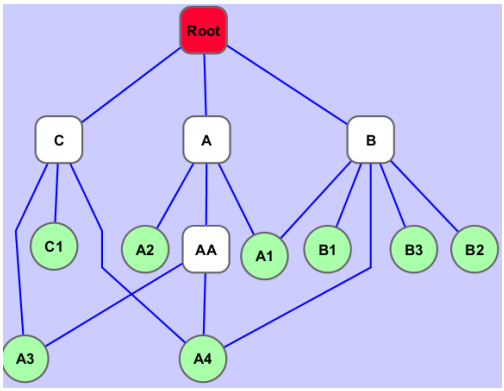


Fig. 3 The tutorial network in Cytoscape contains 13 nodes and 16 edges. The root is represented by red rectangle, other categories are white rectangles and leaf nodes are green circle.

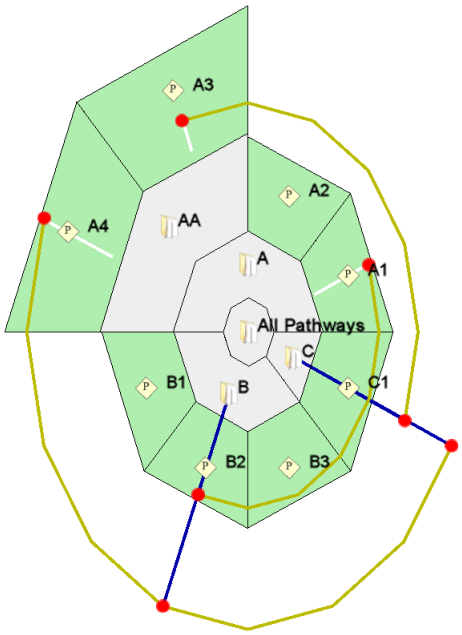


Fig. 4 The same tutorial network in MetNetGE contains 13 nodes and 16 edges.

After finishing the tutorial in Cytoscape, participants were asked to go through the tutorial of MetNetGE. Our hypothesis was that MetNetGE required a greater learning curve than that of Cytoscape. If one user starts the tutorial with Cytoscape may have faster time in learning MetNetGE afterward. We expected that even given this advantage, MetNetGE still requires longer learning time. The networks in the two tutorials are exactly the same, but used completely different metaphors. In order to maintain consistency with the representation in Cytoscape, we didn't use the structure-based coloring [26] or the orbit-based coloring [38] in our study. Instead, we simply colored every leaf node green, and colored every category white. The uplinks are white and the downlinks are blue, and all the orbits are yellow. This simplified color design let participants make connections between this network and the counterpart in Cytoscape, thus making it easier for them to understand the metaphor in MetNetGE.

2.7.2 Understand Topology Structure of Medium Network.

Depending on the ID of the participant, he or she was given the real task on either Cytoscape or MetNetGE. Participants' experiences were counterbalanced, with participants with odd ID numbers starting with Cytoscape. In the Cytoscape version, the medium network for task part 1 is shown in Fig. 5. The top figure shows the overview where individual nodes are hardly visible. As a result, users always need to zoom-in (as in the bottom figure), and focused on a small part of the ontology. Since the nodes and edges are mixed together, participants constantly dragged the nodes to see their edges, and moved nodes to empty spaces. As expected, many participants could not find all the answers of task 5 (selecting pathways which have at least 3 parents) and task 7 (selecting two pairs of highly related categories). They normally gave up after spending 3 - 5 minutes on each task. We counted each missing answer as one error. We will discuss more in the Results section about how we analyzed those missing data points to consider both error and time.

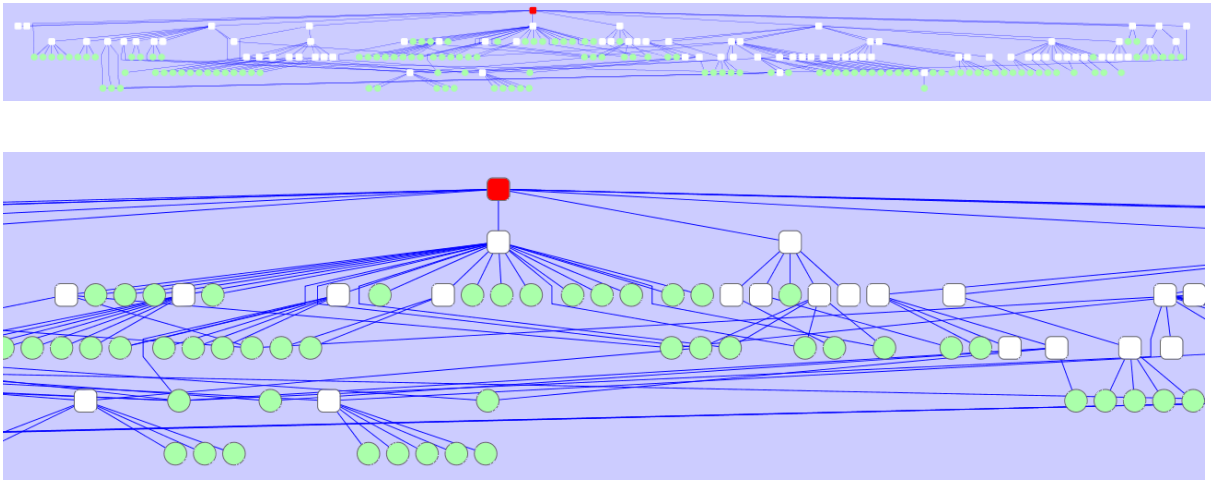


Fig. 5 The medium network in Cytoscape in the user study. Coloring is the same as was used in the tutorial network in the previous figure. (Top) Overview of the full ontology. (Bottom) Zoomed-in view of a small part of the ontology. The network contains 206 nodes and 226 edges.

After finishing task part one with Cytoscape, participants started using MetNetGE to perform task part one. One possible alternative to this procedure is to let participant immediately use Cytoscape to continue part two. However, we found that since the tasks involved too many concepts, it would save overall time if we let user focus on a few concepts at a time, and then move on to new concepts. Since the tasks in parts one and two focus on different concepts, the procedure was designed to relieve participants from keeping too many concepts in mind. Fig. 6 shows the screenshot of viewing the medium size ontology in MetNetGE. Participants can clearly view the multiple inheritance links through the yellow orbits and blue, white links. Finding the highly related categories was initially challenging for participants, because this property is not directly mapped to any metaphor.

To help the user finish this task, we provided hints and reminded participants that the blue link extended from a category shows that that category shares children with another category. This important property is explained in the MetNetGE tutorial, however, few users remembered to use it in this task since it is not used in previous tasks. As a result, a blue link with many red dots means this category shares many children with other category. After this hint, all the participants remembered this property from the tutorial section, and could then find the highly related categories through those blue links. The need for the hint in this task shows that the learning curve for MetNetGE is high, and participants tend to forget its metaphors if not used regularly. For example, Fig. 7 shows how the user can visually find one pair of such highly related categories through blue link and yellow orbits. As soon as participants were reminded of this orbit metaphor, they could quickly find the other pair of highly related categories near the top of the given ontology.

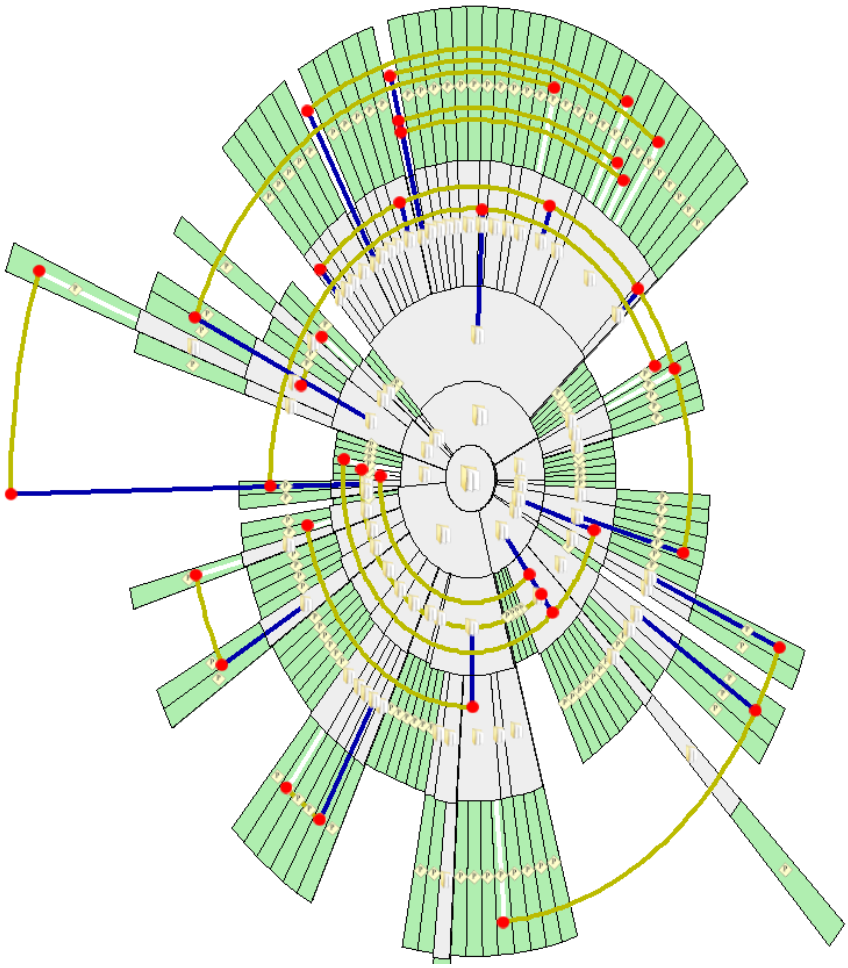


Fig. 6 Medium size ontology is shown in MetNetGE. The ontology is part of the E-Coli pathway ontology, which contains 218 nodes and 238 edges. The color coding is the same as in tutorial network where multiple inheritances are represented by yellow orbits.

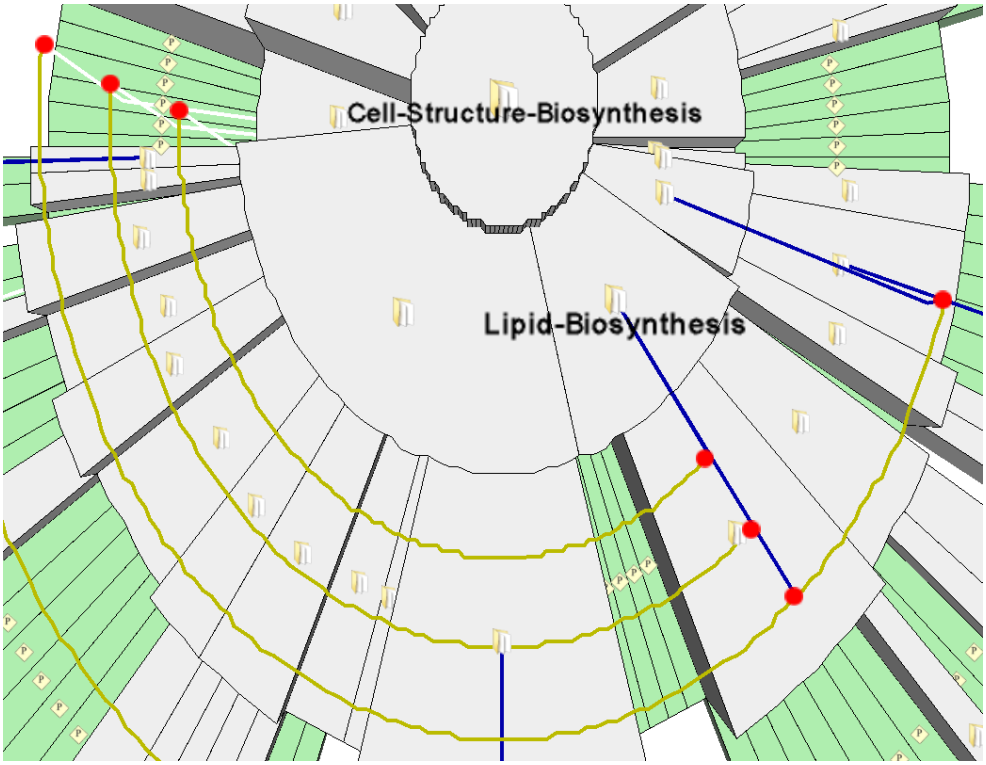


Fig. 7 The zoomed-in view of the medium size ontology. Category Lipid Biosynthesis contains a blue link that intersects three yellow orbits through three red dots. Those three yellow orbits all end in the pathways in the category Cell Structure Biosynthesis. These connections indicate that the two categories are highly related.

2.7.3 Discover Trend of Experimental Data.

After finishing the topology task (part one of the tasks) using both tools, participants moved on to task part two. Again, half of the participants used Cytoscape as their first tool. They were presented with the screen as the one in Fig. 8 where four experimental conditions were mapped to a larger network. The color of each node indicates the average gene expression value in the given condition. Dark green represents a low value while bright red represents a high value (this is the common color coding in biology research). We also prepared an alternative color coding for red-green color blind participants where brown and purple represented the extremes of the color bar. All of the participants were able to distinguish red and green in our user study, and thus we used the normal color coding. As expected, the overview of the ontology in Cytoscape (as in the top of Fig. 8)

becomes very thin and wide, where the color of each individual node can be hardly seen. Participants need to zoom in closely (as in the bottom of Fig. 8) to a small portion of the ontology in order to see the color of nodes and existence of edges clearly.

To switch to other conditions of this gene expression dataset , participants could click on a setting button on the left of Cytoscape's GUI window. The normal workflow in this task was that the participant started with one far end (e.g., the left end) of the ontology and zoomed in such that every node's color could be distinguished. Then he or she switched between conditions to find whether groups of nodes satisfied the task requirements. Normally, participants needed to switch back and forth several times since they could focus only on a small portion of the group at one time. Although the comfortable size of the zoom-in view varied for different participants, the ratio between the visible region of zoom-in view and that of the whole graph typically ranged from 1:4 to 1:5. Therefore participants would repeat the above procedure at least 4 to 5 times to search the answer in the whole graph.

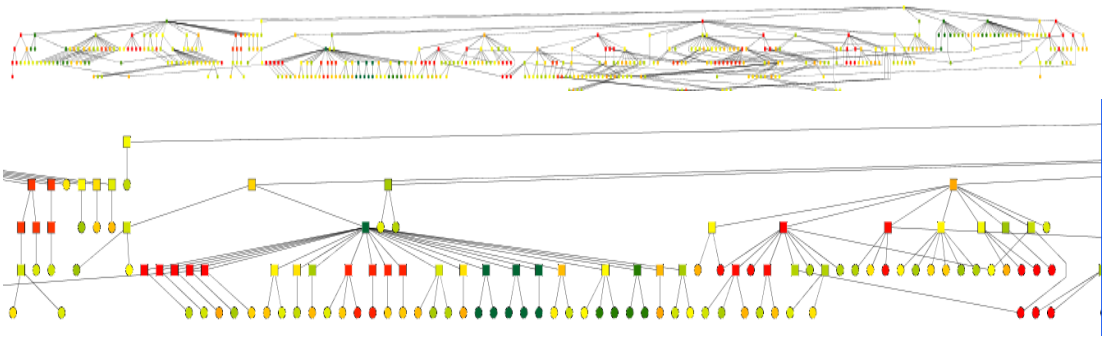


Fig. 8 The large ontology (contains 430 nodes and 457 edges) mapped with omics data in Cytoscape. The top figure shows the overview; the bottom figure shows the zoomed-in view.

Participants also used the MetNetGE tool to view the transcriptomic data. The visualization is shown in Fig. 9. The color coding is the same with the one used in Cytoscape (dark green for low and bright red for high). Since MetNetGE's layout put children directly under their parents, it was easy to see a group of related nodes having

the same color. Also, since participants could always see the whole graph in any of the conditions, they didn't need to always zoom in one part and pan to other parts of the graph. They only need to zoom in when verifying the answers. Most participants found qualified regions by switching conditions only 2-3 times.

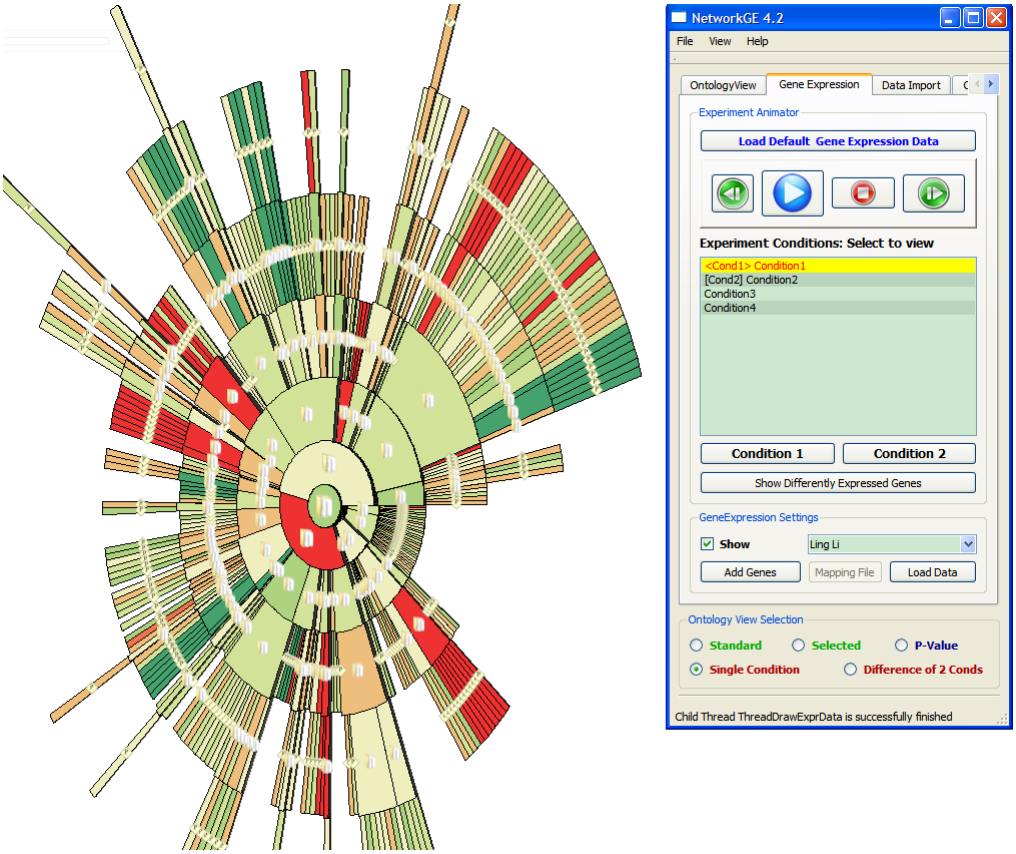


Fig. 9 The whole pathway ontology mapped with the omics data of 4 experimental conditions. Currently, the condition one is shown. Participants can switch conditions in the GUI. The ontology contains 442 nodes and 511 edges.

2.8 Surveys

After participants finished all the tasks, they completed an online post-study survey. No observer was present while participants completed the survey. After submitting the form, participants were finished with the study.

The post-study survey consisted of demographic questions and general questions regarding MetNetGE. The questions are presented in Table 3.

Table 3 The questions used in the post-study survey.

Questions 1 to 5 are demographic, and Questions 6 to 11 focus on evaluation of tools and tasks. Questions 3,4,9 are Likert scale with 5 values.

Demographic Questions.

1 Which department or major are you in?

2 What is your occupation?

3 What's your level of computer skill in terms of using computer software and websites?

4 Are the biological concepts in this user study easy to understand?

5 Your gender?

Evaluation of tools and tasks

6 Which software do you prefer to use to navigate and understand the overview of Pathway Ontology Structure? (Task part one)

7 Which software do you prefer to use viewing experimental conditions on Pathway Ontology? (Task part two)

8 Overall, which software do you enjoy using?

9 Do you feel that the medium and large networks in Cytoscape are more complex than the corresponding ones in MetNetGE?

10 What do you think are the biggest advantages of MetNetGE?

11 What do you think are the primary limitations, disadvantages of MetNetGE?

3 Results

There are 20 participants in our study. As stated in the procedure of the user study, we recorded time duration and number of errors as participants tried to complete each task. It is natural to consider the number of errors as the accuracy of each task. However, since most of the tasks are visualization related, most of the participants can get the correct answers in most cases. The major difference between tools is how long it takes the participants to discover the results visually. We use both completion time and the number of errors as indicators of the efficiency of each tool.

We divided the tasks into meaningful groups and record time for each group. For example, we make Questions 1 to 3 to one group which focused on the overall topological knowledge of the medium size ontology. Questions 4 to 6 are grouped together since they both concerned with the multiple inheritance. Question 7 is focused on the highly related categories, and used much more time than any other single question. Therefore it is considered as a single task group. All three questions in part two are focused on discovering interesting trend in experimental conditions, thus are considered one group.

3.1 Task completion time

Fig. 10 shows the boxplot of completion time for the tutorial tasks. As expected, participants took significantly more time to learn MetNetGE (*n=20, p < .001)*. The difference in average time is 330 seconds (5.5 minutes). The reason is that Cytoscape used the traditional node and link graph to represent ontology, which is a familiar way to investigate ontologies.

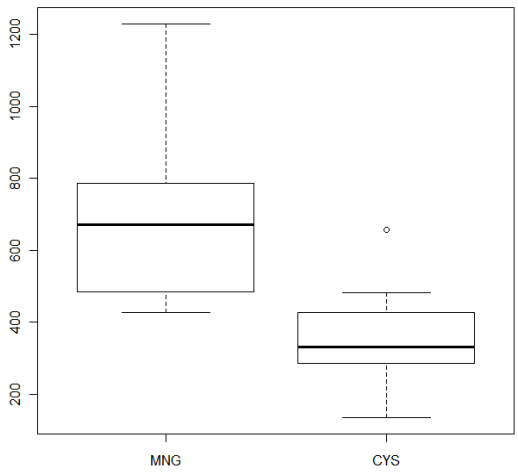


Fig. 10 Boxplot of the completion time for tutorial task. The lower bar shows the minimum. The lower boundary of the box shows the 25%, or lower quartile. The bar inside the box shows the median. The upper boundary of the box represents the 75%, or upper quartile. The upper bar shows the maximum. Finally, the small circles represent the outliers.

We grouped similar tasks and analyzed the completion time of these task groups to see if there is difference in using different tools. One thing to notice is that several participants didn't correctly answer question Q6 and Q7. E.g., some participants could find only one pair of highly related categories in Cytoscape. In this case, we considered it as a missing data problem. As a result, for participants who had one error, the completion time for Q7 in Cytoscape is the actual recorded time plus average time to get one correct answer (it is around 150 seconds in our study)

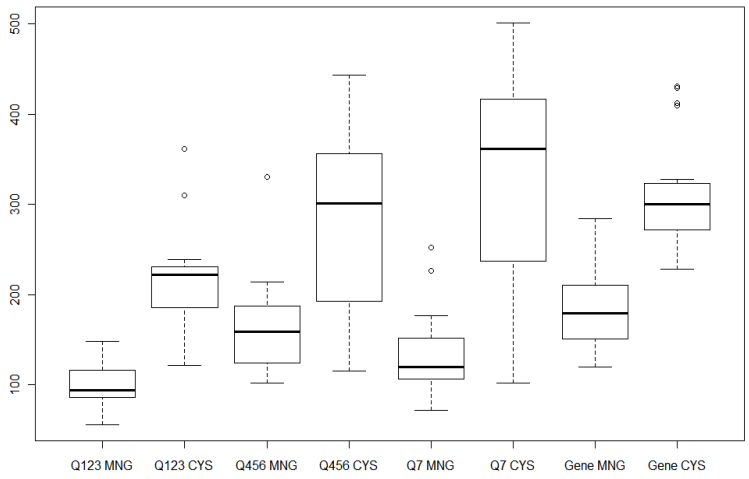


Fig. 11 Boxplot of completion time of all task groups.

As we can see from Fig. 11, for all the task groups, MetNetGE has lower mean task completion times than Cytoscape. Typically users completed tasks using MetNetGE twice as fast as when using Cytoscape.

3.2 Normality test of sample data

Given the within subjects design, we used a dependent t-test for paired samples. One common assumption for the sample data of t-test is that the sample dataset should follow a normal distribution [44]. We tested the normal distribution of the user study dataset with the Shapiro-Wilk Normality Test [45, 46]. We used the Shapiro.test() function from

R, and got the result in **Table 5** in the Appendix. The higher the p-value, the more likely the sample data follows the normal distribution. If the p-value is greater than 0.05, we can say the sample data are fit the normal. Since most of the recorded times follow the normal distribution, they are good candidates for Student's t-test.

It is also worth noticing that the time to finish the tutorial of MetNetGE barely follows the Normal distribution (p-value 0.0526). One possible reason for this is participants involve both native and non-native English speakers. Since the reading material in tutorial section is very long (more than 500 words), native English speakers have clear advantages in reading and going through the tutorial much faster. The other possible reason is that some biologists may be more familiar with the concept of ontology than some computer science students, thus making them progress through the tutorial more easily.

The other interesting result is the completion time for question 7 of MetNetGE didn't follow the normal distribution (p-value 0.0046). One possible explanation is that to find the pair of highly related categories most quickly, participants needed to find the blue downlinks which intersect many orbits, and then trace those orbits to confirm that at least three of them are originating from the same category. This workflow is the reverse order of how they would find multiple inheritance links (e.g. find pathways that have at least 3 parents). These data indicate that participants may not have equal ability to think creatively and get the right answer by reversing the normal workflow.

3.3 Statistical analysis of results

The result of the Student's t-test is shown in **Table 4** and all p-values are much less than 0.01. Some possible explanations are listed in the Discussion section.

Table 4 Student's T-Test results of all sample data.

The null hypothesis is that the mean completion times for MetNetGE and Cytoscape task groups are the same. P-value <= 0.05 means the Null Hypothesis is rejected, and there is statistically significant difference between MetNetGE and Cytoscape in terms of completion time.

Sample Data	**T**	**df**	**p-value**
Tutorial Completion Time in seconds	5.73	28.19	3.64E-6
Q123 Completion Time	-8.6818	26.267	3.39E-9
Q456 Completion Time	-4.4975	28.739	1.04E-4
Q7 Completion Time	-7.1657	24.933	1.67E-7
Gene Expression Data Completion Time	-7.1874	35.162	2.14E-8

3.4 User preference and comments

Beside the objective measurements (completion time and number of errors) we gathered users' subjective opinions after using both tools. Table 3 listed the questions used.

Q5 asked participants to choose which tool they preferred using to view the overview of pathway ontology structure. As we can see from Fig. 12, over 80% of participants selected MetNetGE, 2 participants selected "Hard to say", and only 1 participant select Cytoscape.

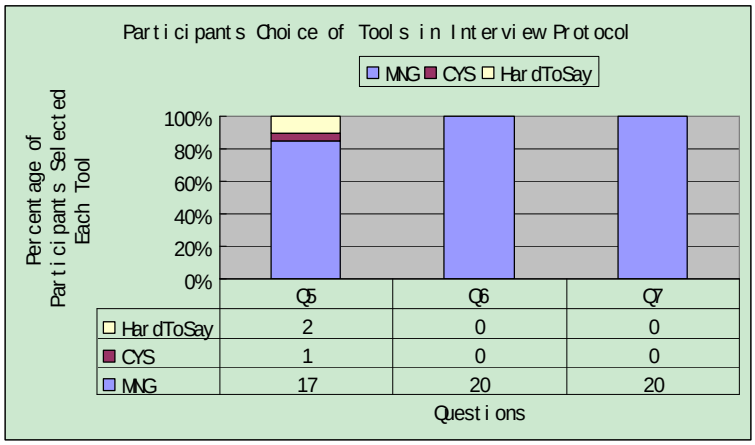


Fig. 12 Percentage and count of how many participants have chosen each tool in questions 6 to 8 in post-study survey. MNG stands for MetNetGE, CYS stands for Cytoscape.

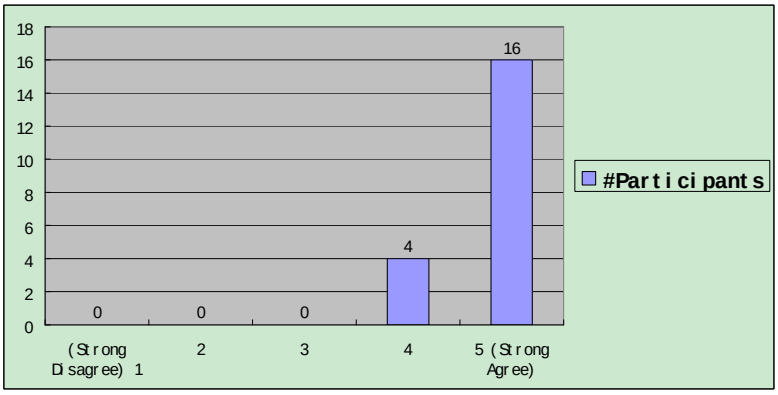


Fig. 13 Percentage and count of how many participants agree that "network in Cytoscape is more complex than MetNetGE" (Question 8). The question is presented as a 1 to 5 scale, where 5 means strongly agree.

One interesting and expected finding is that, 80% of participants strongly agree that the network in Cytoscape is more complex than the one in MetNetGE. In fact, the network in MetNetGE is slightly more complex than the one in Cytoscape in terms of number of nodes and edges (218 nodes and 238 edges vs. 206 nodes and 226 edges). The major reason for this result is that Cytoscape's node and link representation contains many edge crossing, which is considered one of the most important metrics in causing visual complexity [23] in graph layout algorithms. Although fewer edge crossings do not guarantee better layout, more edge crossings hurt the aesthetics and presentation of graphs.

Questions 10 and 11 invited participants to write comments about the advantages and disadvantages of MetNetGE regarding the tasks. The common advantages participants wrote are “easy to view conditions of experiments,” “utilize screen real-estate much better,” “no edge overlapping,” and “all the data is prominently visible.” Some comments are below.

- “*The layout utilizes the screen real-estate much better than Cytoscape, so all the data is prominently visible. Also, navigation around network feels much easier to comprehend due to the various visual elements that clearly stand out.”*
- *“The user can easily grasp the rough idea about the whole network in a very efficient way, e.g. depth of the ontology, # of categories, etc.”*
- *“It creates a very pleasant and user-friendly environment for the user to play with it, and try out its functionalities.”*
- “*The concentric ring layout makes it very easy to see the direct children and descendants of a category. This is a large advantage over trying to follow lines in Cytoscape. ”*

Although most of the disadvantages are about the high learning curve, participants also mentioned that after getting used to it, the tasks are easier to finish. Some quotes are listed below.

- *“While I find the poles linking a child to multiple parents to be confusing at first, I ultimately found it easier to use in this user study.”*
- *“Need more time to understand the terminology and rules, but once I get familiar with them, it becomes easier later”*

Participants who did not choose MetNetGE as their preferred tool to use in tasks about pathway ontology structure noted that the metaphor was not straightforward and required a high learning curve. For example,

- *“It takes some getting used to since it is a new way of representing data. But once that is done, I guess its way better than using Cytoscape for the same purpose.”*

The other common complaint is that the layout is static where participants can’t move the nodes and regions around, a major limitation of MetNetGE. For example,

- *“Not being able to move items around limits the user to only a single view of the data. While the view is a good one, in Cytoscape I can "filter" out wrong answers when I discover them by moving the particular node off to the side.”*

4 Discussion

4.1 Mental model and learning curve.

One possible reason for the high learning curve of MetNetGE is that the radial space filling view of the ontology doesn’t fit the mental model of common participants.

Mental models were first introduced by Johnson-Laird [47, 48] as an internal mental representation of something in the world. Norman and Payne [49, 50] then modified and extended the models to adapt new research studies and discoveries. Two types of mental models are defined and well studied [51]: structural and functional. Generally, a structural model makes predictions of actions based on facts about a system, while a functional model describes actions that the system should take under specific circumstances.

Ontologies are widely studied and used in many areas, e.g. company hierarchy, family relations, computer diagrams. In most visualization in those areas, ontologies are represented as node-link graphs. As a result, participants have formed the structural mental model that the ontology is a graph with many nodes and edges connecting nodes. Therefore, they could easily understand Cytoscape’s node-link diagram for the small tutorial network. Switching to the MetNetGE’s radial space filling (RSF) layout contradicted most participants' own structural mental model of how an ontology should look. Thus, they needed more time to first resolve the contradiction between their existing mental model and the presented system image, and then to form the new mental model.

Evidence of this contradiction was observed when we introduced MetNetGE’s orbit metaphor to represent multiple inheritances. Participants were first given instruction on radial space filling without orbits. Then after the orbits were introduced, some

participants grew confused about which were the categories (parents) and which were the leaf nodes (children). This may be due to the graphical similarity of orbits in MetNetGE to edges in Cytoscape (they are lines), while edges in Cytoscape always connect nodes, which is not always true in MetNetGE.

After participants correctly understood the tasks in tutorial section, most of them formed the new mental model for MetNetGE and used it correctly to finish later tasks more quickly.

4.2 Analysis of tasks related to multi-inheritance.

As shown in Fig. 10 and Fig. 11, although MetNetGE users required much longer than Cytoscape users to finish their respective tutorials, MetNetGE had statistically better completion times for our selected tasks and stronger post-task user preferences. The main reason for the advantage of MetNetGE in topological tasks is that MetNetGE directly and clearly represents this information by its ERSF (enhanced radial space filling) layout. For example, Question 2 asks to find the category in level 2 which may contain the most pathways. Although neither tool can give an exact answer to this question, MetNetGE's ERSF draws the angle of each category proportionally to how many leaf nodes exist under its spanning tree, which is a good approximation of how many pathways it contains. As a result, participants need only to scan through all categories in level 2, see which has the largest angle, and choose the clear winner, the category IND-AMINO-ACID-SYN (short for Individual Amino Acids Biosynthesis). The result is shown in Fig. 14.

To answer this question in Cytoscape, participants need to dig into the whole complex drawing where all the ontology relationships were represented by edges (as in Fig. 5). One strategy commonly deployed by participants was to first identify candidate

categories in level 2, and then drag them to the top of the graph. Then they scanned the tree under each category to find the largest one. This is a very time-consuming task, and participants normally spend more time to verify their result than while using MetNetGE, which indicates they are less confident about their answer with Cytoscape.

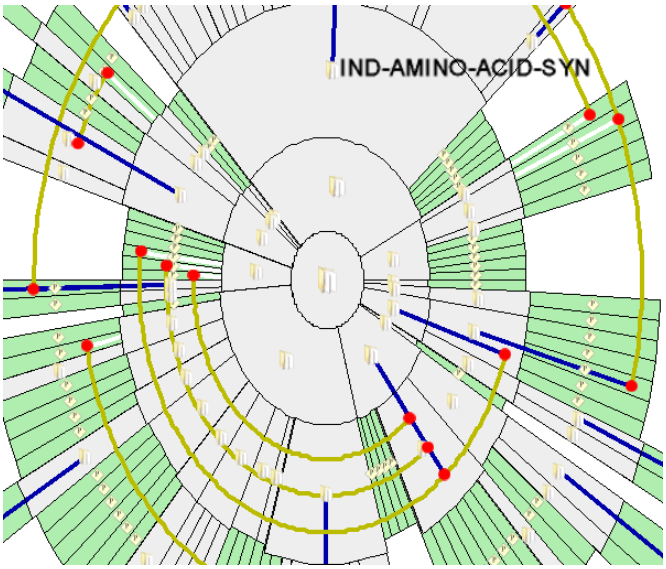


Fig. 14 The view of medium sized ontology in MetNetGE. To find the category in level 2 which contains the most pathways, participants need only to find the category with largest angle.

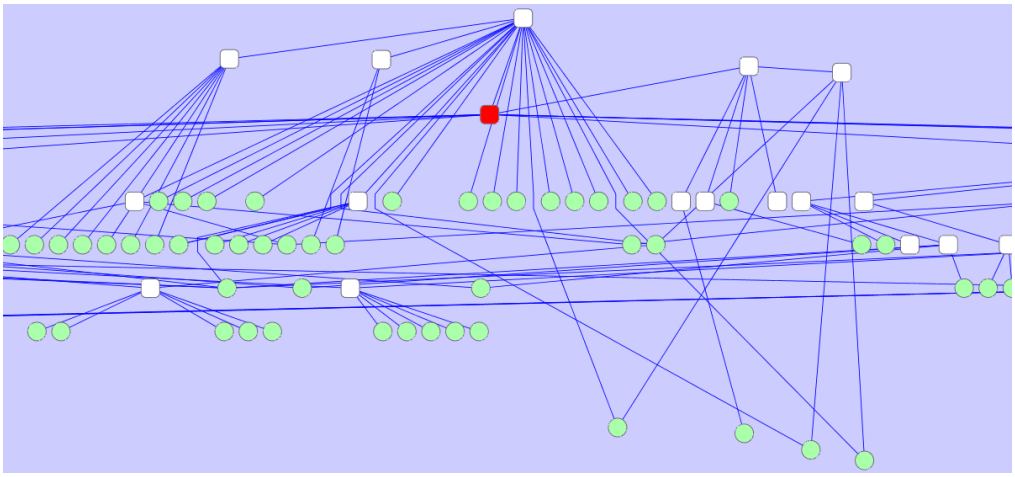


Fig. 15 This layout is modified by the user to finish tasks. To find the highly-related categories, many users moved candidate categories to the top of the drawing, and moved their children to the bottom.

Fig. 14 also shows that MetNetGE has a clear advantage in completing multiple inheritances related tasks in questions 4, 5 and 6, e.g., finding pathways that have at least 3 parents. On average, as shown in Fig. 11, participants using MetNetGE required about

half the time to find answers to these questions than in Cytoscape. The main reason for this difference is likely that MetNetGE's orbit metaphor clearly manifested this information. To find the pathway which had at least 3 parents, participants needed only to iterate through all the orbits, since only pathways with multiple parents can have orbits. They then scanned those orbits to find the ones having at least 3 red dots, where each red dot means the pathway has one parent. However, when using Cytoscape, participants needed to go through almost every node. Since all the nodes are linked with edges, it's not clear which nodes have more than one edge when the edges are all cluttered together as in Fig. 5. Participants needed to drag nodes somewhere with extra space to see the edges incident to them. Fig. 15 shows the layout after one participant had finished the task. We can see that the final layout looks quite different than the original, which means the participant modified the graph intensively. As a result, the time to complete this task in Cytoscape is proportional to the number of nodes in the graph, which can be represented as $O(n)$, if n is the number of nodes. With MetNetGE's orbit metaphor, the time to complete this task can be shortened to $O(k)$ where k is the number of pathways with multiple parents and $k<<n$.

The above difference in completion time can also be explained by the Gestalt law of perceptual organization [52]. Gestalt is a psychology term which means "unified whole" [53]. Gestalt theory attempts to describe how people organize visual elements into perceptual groups or unified wholes when certain principles are applied.

One of the important grouping principles in Gestalt theory is similarity vs. anomaly. Anomaly occurs when an object is emphasized because it is dissimilar to the objects around it. Gestalt law states that the dissimilar objects normally become the focal points and get more attention. In Cytoscape, all the relations and connections are represented as

links, thus the nodes which contain more parents are similar to other nodes. In MetNetGE, the nodes with single parents don’t have any links or orbits. Thus the nodes with multiple parents (having orbits) become anomalous and get users’ attention easily. This difference is likely one of the major reasons why participants can find such nodes much more quickly in MetNetGE.

It is also interesting to see that participants using Cytoscape have a larger variation in completion time. In Fig. 11, the standard deviation and size of box plot of Cytoscape is twice as large as that of MetNetGE. We suggest the reason is that since participants needed to drag and investigate almost every node to find the answer, some participants were fortunate to select the right nodes after investigating only a couple of nodes. Some participants who were not as fortunate may have tried to first untangle the nodes in the densest area, and it turned out that those nodes had only one or two parents.

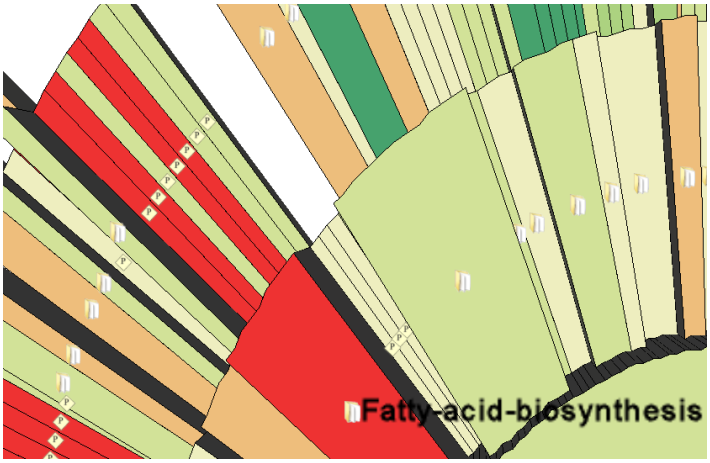
In the post-study questionnaire, several users pointed out that MetNetGE lacks the ability to interactively move the nodes and regions around. This requirement shows that letting user modify the graph to create better views is an important feature for visualization tools.

4.3 Analysis of tasks related to gene expression experimental data

In the part two of our study, the tasks are related to understanding the gene expression dataset on the pathway ontology. Participants are asked to find which parts of the ontology are highly or differentially expressed in given experimental conditions. In general, the tasks for understanding gene expression data are easier than the tasks to understand ontology structure. We observed that even in Cytoscape, participants normally didn't need to rearrange the nodes or modify the graph structure. However, MetNetGE

was still much more efficient than Cytoscape in terms of completion time. We can explain this by the Gestalt laws of closure and proximity [54].

Proximity occurs when elements are placed close together. Participants tended to perceive those elements as a group. The experimental task deliberately asked participants to find a region of nodes that consisted of one category and at least three of its children. In MetNetGE, most children of a category are placed immediately around the edge of the category itself, thus they are close together and participants naturally perceived them as a visual group, or region. On the contrary, Cytoscape may place some children far away from their parents, thus making it difficult for participants to realize those nodes formed a region. A lot of participants' time was spent on examining the edges between nodes to verify whether they were in the same region. Fig. 16 shows one example of representing a region in both MetNetGE and Cytoscape. It is clear that it requires much more effort to realize the red nodes in Cytoscape formed a region.



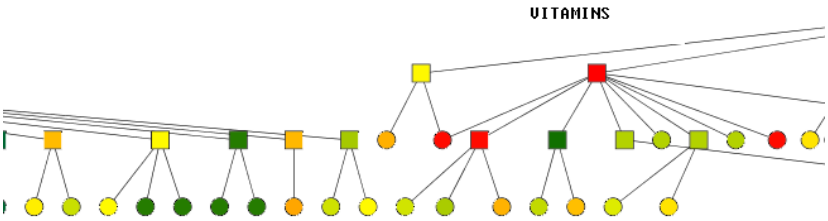


Fig. 16 Different representations of one region. MetNetGE (top) shows the region (Fatty acid biosynthesis) as red blocks placed together. Cytoscape (bottom) shows the region (Vitamins) as red nodes connected by edges where the children may be far away from their parent.

The other possible reason for the advantages of MetNetGE is the effective use of screen space by the ERSF layout, which was also pointed out by several participants. In Cytoscape, a node's color represents its omics value. However, a vast amount of screen space is blank, and the colored region is relatively small compared with the whole screen. When participants zoomed out in Cytoscape to view the whole graph, the color and connections of nodes or regions grow indistinguishable. Thus, participants always need to zoom in to focus on a small part of the group and then pan to other parts. These extra actions created unnecessary discontinuities in the workflow, introducing an extra burden to participants. In MetNetGE, almost all screen space is utilized to show data as colored blocks. Participants can therefore comfortably zoom out to see the whole screen while maintaining the visibility of individual region (as in Fig. 17). The workflow to examine the experimental value remains continuous.

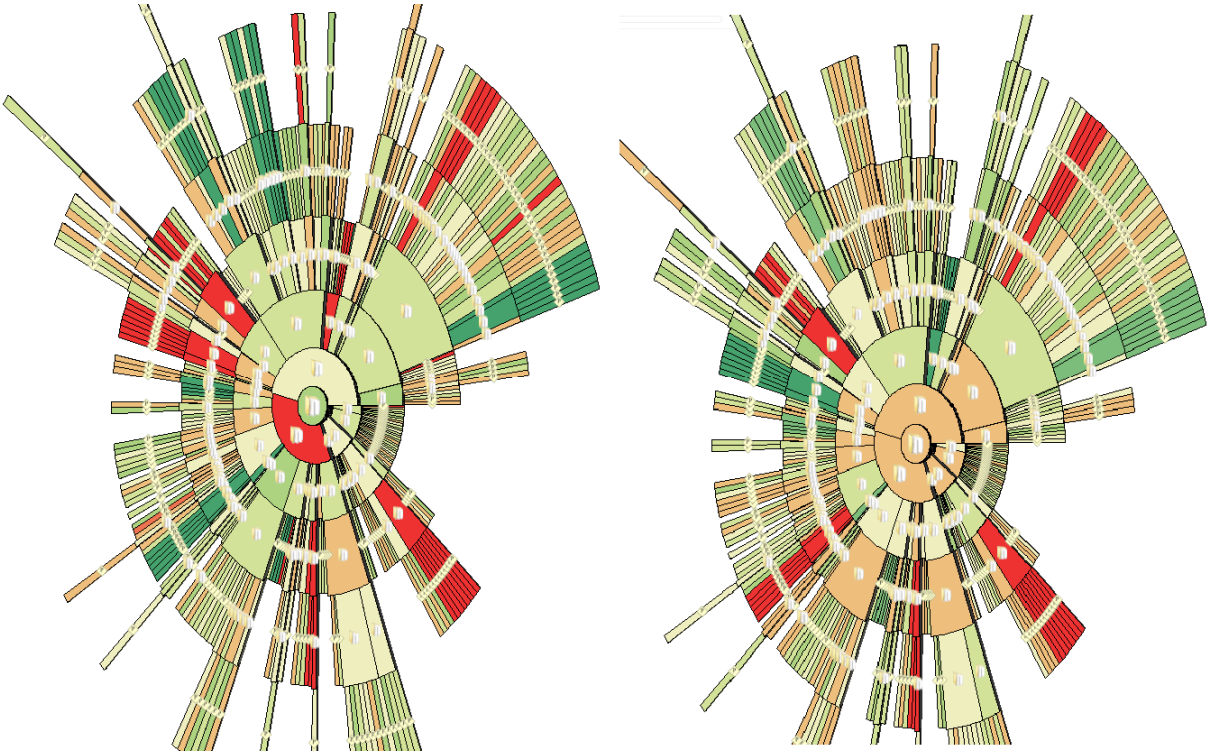


Fig. 17 Screenshot of two experimental conditions in MetNetGE. Participants can easily compare and find regions that changed color in these conditions in the whole graph.

5 Conclusion

Linking large-volume experimental data with hierarchical ontologies that relate biological concepts is a key step for understanding complex biological systems. The visualization of these data needs to clearly represent the non-tree edges in the ontology structure and present the whole experimental data in one screen for biologists to understand the overall effects of the experiments. However current visualize tools lack the above abilities. The authors have proposed the radial space-filling (ERSF) algorithm [37] to meet all the visualization requirements. In this paper, we reported the procedures and results of one user study to compare the ERSF and MetNetGE with a widely known software tool.

On average, participants of the user study took about twice the time to finish the tutorial section of MetNetGE as compared to Cytoscape. However, when working on the

real tasks, participants used only about half the time in MetNetGE. For all the task groups, the performance in using MetNetGE is statistically significant better than that of Cytoscape. In conclusion, our user study clearly demonstrates that the ERSF algorithm provides biologists more efficient ways to visualize and analyze ontology and pathway data.

Acknowledgements

This work is supported by NSF grant #IIS0612240. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We are grateful to Dr. Li Ling for valuable biological input during development of this tool.

Appendix

Table 5 The Normality test result for all sample data (recorded time for each task group).

For simplicity, MNG stands for MetNetGE, CYS stands for Cytoscape. The p-value with green color means the sample passed the normality test, thus it follows the normal distribution. The p-value with yellow color means the sample barely passed the normality test. Red means the sample is not following normal distribution.

Sample Data	Sample Size	pValue
Tutorial MNG	20	0.0526
Tutorial CYS	20	0.3122
Q123 MNG	20	0.834
Q123 CYS	20	0.092
Q456 MNG	20	0.112
Q456 CYS	20	0.261

Q7 MNG	20	0.004635
Q7 CYS	20	0.332
Experimental MNG	20	0.264
Experimental CYS	20	0.067

References

1. Keseler, I.M., et al., *EcoCyc: a comprehensive view of Escherichia coli biology*. Nucleic Acids Res, 2009. **37**(Database issue): p. D464-70.
2. Carbon, S., et al., *AmiGO: online access to ontology and annotation data.* Bioinformatics, 2009. **25**(2): p. 288-9.
3. Day-Richter, J., et al., *OBO-Edit--an ontology editor for biologists.* Bioinformatics, 2007. **23**(16): p. 2198-200.
4. Maere, S., K. Heymans, and M. Kuiper, *BiNGO: a Cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks.* Bioinformatics, 2005. **21**(16): p. 3448-9.
5. Jia, M., et al., *Visualizing Multivariate Hierarchic Data Using Enhanced Radial Space-Filling Layout*, in *Advances in Visual Computing*. 2010, Springer Berlin / Heidelberg. p. 350-360.
6. Jia, M., et al., *MetNetGE: interactive views of biological networks and ontologies.* BMC Bioinformatics, 2010. **11**(1): p. 469.
7. Katifori, A., et al., *Ontology visualization methods a survey*. ACM Computing Surveys, 2007. **39**(4): p. 10.
8. Green, M.L. and P.D. Karp, *The outcomes of pathway database computations depend on pathway ontology*. Nucleic Acids Res, 2006. **34**(13): p. 3687-97.
9. Katifori, A., et al. *Selected results of a comparative study of four ontology visualization methods for information retrieval tasks*. in *Research Challenges in Information Science, 2008. RCIS 2008. Second International Conference on*. 2008.
10. Ellson, J., E.R. Gansner, and E. Koutsofios, *Graphviz and dynagraph static and dynamic graph drawing tools.* 2003, Technical report, AT&T Labs - Research.
11. Baehrecke, E.H., et al., *Visualization and analysis of microarray and gene ontology data with treemaps.* BMC Bioinformatics, 2004. **5**: p. 84.
12. Tekusova, T. and T. Schreck. *Visualizing Time-Dependent Data in Multivariate Hierarchic Plots -Design and Evaluation of an Economic Application*. in *Information Visualisation, 2008. IV '08.* 2008. Columbus, OHIO, USA.
13. Munzner, T., *Exploring Large Graphs in 3D Hyperbolic Space.* IEEE Computer Graphics and Applications, 1998. **18**(4): p. 18-23.
14. John, S., *An evaluation of space-filling information visualizations for depicting hierarchical structures*. 2000, Academic Press, Inc. p. 663-694.
15. Yang, J., et al., *InterRing: a visual interface for navigating and manipulating hierarchies*. 2003, Palgrave Macmillan. p. 16-30.
16. Consortium, G.O. *GO Slim and Subset Guide*. 2009 [cited; Available from: http://www.geneontology.org/GO.slims.shtml.
17. Dwyer, T., et al., *A comparison of user-generated and automatic graph layouts.* IEEE Trans Vis Comput Graph, 2009. **15**(6): p. 961-8.
18. The-R-Foundation. *Introduction to R*. 2011 [cited; Available from: http://www.r-project.org/.
19. Mankiewicz, R., *The Story of Mathematics*. 2004, Princeton University Press. p. 158.
20. Wiki. *Dependent t-test for paired samples*. 2011 [cited; Available from: http://en.wikipedia.org/wiki/Student's_t-test#Dependent_t-test_for_paired_samples.
21. Casella, G.B., Roger L., *Statistical inference*. 2nd ed. 2001: Duxbury.
22. Shapiro, S.S.W., M. B., *An analysis of variance test for normality (complete samples).* Biometrika, 1965. **52**(3-4): p. 591–611.
23. Wiki. *Shapiro–Wilk test*. 2011 [cited; Available from: http://en.wikipedia.org/wiki/Shapiro%E2%80%93Wilk_test.
24. Johnson-Laird, P.N., *Mental models and human reasoning.* Proc Natl Acad Sci U S A. **107**(43): p. 18243-50.
25. Johnson-Laird, P.N., V. Girotto, and P. Legrenzi, *Reasoning from inconsistency to consistency.* Psychol Rev, 2004. **111**(3): p. 640-61.
26. Norman, D.A., *Some observations on mental models*, in *Human-computer interaction*, R.M. Baecker, Editor. 1987, Morgan Kaufmann Publishers Inc. p. 241-244.
27. Payne, S.J., *Users' Mental Models: The Very Ideas.* HCI Models, Theories, and Frameworks, 2003: p. 135-154.
28. Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S.,, *Human-Computer Interaction*. 1994: Addison-Wesley.
29. Koffka, K., *Principles of Gestalt Psychology*. 1935, New York: Harcourt Brace.

30. *The Gestalt Principles*. 2011 [cited; Available from: http://graphicdesign.spokanefalls.edu/tutorials/process/gestaltprinciples/gestaltprinc.htm.

Chapter 9. Conclusions

9.1. Summary

Meaningful visualization of large scale biological data is the key for achieving new discoveries in system biology research. However, visualization tools used in these areas often fail to present a meaningful and insightful view of underlining data.

MetNetGE features a novel approach to integrate the visualization of three different datasets: the pathway diagrams, pathway ontology, and the omics data. We organized the pathway diagrams by pathway ontology and proposed the Enhanced Radial Space-Filling (ERSF) technique to layout and show this ontology. Each ontology node is represented as a colorful region in the drawing, and the detailed pathway diagram is drawn inside the region. The multiple inheritance relationship is represented by the concept of “orbits”. This technique can show the structure of ontology with hundreds of nodes in one computer screen, and facilitate the user to trace the non-tree edges which may represent interesting relations.

For a detailed view of individual pathways, the 3D tiered layout can be used to group nodes into distinct layers based on node type or sub-cellular location. Instead of generating layouts for each layer independently, we first calculate nodes’ positions on one major plane, and then compute other nodes gradually. This layout helps review cross-layer patterns as well as letting the main metabolic reactions standout.

The omics data were mapped onto both the ontology drawing and the pathway diagram. By mapping average expression values, differentially expressed genes and statistical test results onto the ontology drawing, MetNetGE enables biologists to discover interesting patterns at a larger scale.

To demonstrate the effectiveness of our proposed algorithms, we conducted a user study with 20 participants. The user study let participants to use MetNetGE and Cytoscape to complete several biological tasks. The tasks were selected as abstraction of tasks biologist performed in day to day work. The completion time for each task and each tool were recorded and analyzed. Although MetNetGE requires higher learning time (680 seconds vs. 350 seconds) on average, it helps participants quickly finish the tasks. For all the tasks, participants used significantly less time in MetNetGE than in Cytoscape. For example, tasks for finding ontology terms with multiple parents is 164 vs. 227 seconds; finding highly related categories is 133 vs. 324 seconds; and finding important region of gene expression data is 186 vs. 311 seconds. Besides the objective measurement of efficiency in completing tasks, more than 80% of participants selected MetNetGE as their preferred tool for completing ontology tasks and all participants prefer using MetNetGE for gene expression tasks.

The main advantage of ERSF layout is the efficient use of screen space. One dataset with around 500 nodes in the ontology can fit nicely within one screen while each individual node is distinguishable. The other advantage of ERSF is using regions to represent main hierarchy and using links to represent multiple inheritance relationship, which are often the interesting part in the ontology. This technique makes the multiple inheritance relationship to be easily identified. One major disadvantage of ERSF is the high learning curve, since users are not used to represent ontology in spatial layout. We implemented the ERSF in MetNetGE using Google Earth API. This implementation can allow us to quickly build prototype to demonstrate the effectiveness of ERSF. However, the limitation of Google Earth API also constraint us to only generating static drawings, instead of dynamically changed graphs as used in Cytoscape.

The Aligned 3D tiered layout can help user quickly understand the structure of fairly complex biological pathways (around 100 nodes). However, since it is also implemented using Google Earth API, the drawing is static.

9.2. Future Work

The ability to allow participants to manipulate or modify the graph in real-time is highly requested. Several participants suggested in the post-study interview that MetNetGE should enable users to move and modify the ontology as they worked. Unfortunately, due to the limitation of Google Earth API, programmers didn't have the ability to modify the graph shown in GE programmatically. Thus, we did not modify many important features of the graph on the fly, e.g., changing spanning angle, removing ontology nodes using animation which would lead to future studies.

Due to the high complexity and large scale of biological data, users don't want to view all the data and details simultaneously. They want to see the overview information first, navigate the interesting part, and see details on demand. As a result, a fully interactive system, like Cytoscape, is much better for biologists users in general tasks. However, Cytoscape is limited by its ability that can mainly handle node link representations. Future work which substitutes new graphical engines may allow plug-ins to draw 3D space filling graphs.

MetNetGE can let users navigate the structure of one ontology and map data on it. It would be interesting to also see multiple ontologies together. For example, the GOslim of Arabidopsis may be slightly different than that of *E.coli*. Visualization tools can overlap these two medium sized ontologies together and then the difference of these ontologies can be clearly represented. Although, MetNetGE can't implement this feature, new 3D

visualization platforms may implement this feature using techniques similar to the two and half dimension layout [19].

Availability and requirements

Project name: MetNetGE

Project home page: http://www.metnetge.org

Operating systems: Windows

Programming language: Python

Other requirements: Python 2.5 or higher; Google Earth, PyQt and other required libraries (listed in the documentation on project home page)

License: Freely available under GNU GPL.

Restrictions to use by non-academics: None

Acknowledgements

This work is supported by NSF grant #IIS0612240. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We would like to thank Muhieddine Kaissi for insightful discussions on software design. We are grateful to Drs. Li Ling and Dr. Siva Swaminathan for valuable biological input during development of this tool. We would also like to thank Dr. Ming Zhou for the suggestion on statistical methods in the analysis of user study result.

References

[1] G. D. Bader, D. Betel, and C. W. Hogue, "BIND: the Biomolecular Interaction Network Database," *Nucleic Acids Res*, vol. 31, pp. 248-50, Jan 1 2003.

[2] P. D. Karp, C. A. Ouzounis, C. Moore-Kochlacs, et al., "Expansion of the BioCyc collection of pathway/genome databases to 160 genomes," *Nucleic Acids Res*, vol. 33, pp. 6083-9, 2005.

[3] E. Wurtele, Li, L., Berleant, D., Cook, D., Dickerson, J., Ding, J., Hofmann, H., Lawrence, M., Lee, E., Li, J., Mentzen, W., Miller, L., Nikolau, B., Ransom, N. and Wang, Y, "MetNet: Systems Biology Software for Arabidopsis.," in *Concepts in Plant Metabolomics*, Springer Verlag, 2007, pp. 145-158.

[4] M. Kanehisa, S. Goto, S. Kawashima, et al., "The KEGG resource for deciphering the genome," *Nucl. Acids Res.*, vol. 32, pp. D277-280, January 1, 2004 2004.

[5] P. Shannon, A. Markiel, O. Ozier, et al., "Cytoscape: a software environment for integrated models of biomolecular interaction networks," *Genome Res*, vol. 13, pp. 2498-504, Nov 2003.

[6] Z. Hu, D. M. Ng, T. Yamada, et al., "VisANT 3.0: new modules for pathway visualization, editing, prediction and construction," *Nucleic Acids Res*, vol. 35, pp. W625-32, Jul 2007.

[7] M. Suderman and M. Hallett, "Tools for visually exploring biological networks," *Bioinformatics*, vol. 23, pp. 2651-9, Oct 15 2007.

[8] E. P. Solomon, L. R. Berg, and D. W. Martin, *Biology*. St. Paul, Minnesota, Brooks Cole Pub Co,2008.

[9] M. L. Green and P. D. Karp, "The outcomes of pathway database computations depend on pathway ontology," *Nucleic Acids Res*, vol. 34, pp. 3687-97, 2006.

[10] Gene Ontology Consortium, *GeneOntology*, www.geneontology.org, 2011.

[11] I. M. Keseler, C. Bonavides-Martinez, J. Collado-Vides, et al., "EcoCyc: a comprehensive view of Escherichia coli biology," *Nucleic Acids Res*, vol. 37, pp. D464-70, Jan 2009.

[12] T. Munzner, "Process and Pitfalls in Writing Information Visualization Research Papers," in *Information Visualization: Human-Centered Issues and Perspectives* New York, NY, Springer-Verlag, 2008, pp. 134-153.

[13] J. Day-Richter, M. A. Harris, M. Haendel, et al., "OBO-Edit--an ontology editor for biologists," *Bioinformatics*, vol. 23, pp. 2198-200, Aug 15 2007.

[14] A. Barsky, T. Munzner, J. Gardy, et al., "Cerebral: Visualizing Multiple Experimental Conditions on a Graph with Biological Context," in *InfoVis*, 2008, pp. 1253-1260.

[15] E. Baehrecke, N. Dang, K. Babaria, et al., "Visualization and analysis of microarray and gene ontology data with treemaps." vol. 5, 2004, p. 84.

[16] M. Kanehisa, M. Araki, S. Goto, et al., "KEGG for linking genomes to life and the environment," *Nucleic Acids Res*, vol. 36, Jan 2008.

[17] I. Rojdestvenski, "Metabolic pathways in three dimensions," *Bioinformatics*, vol. 19, pp. 2436-41, Dec 12 2003.

[18] Y. Yang, L. Engin, E. S. Wurtele, et al., "Integration of metabolic networks and gene expression in virtual reality," *Bioinformatics*, vol. 21, pp. 3645-50, Sep 15 2005.

[19] U. Brandes, T. Dwyer, and F. Schreiber, "Visual Understanding of Metabolic Pathways across Organisms Using Layout in Two and a Half Dimensions.," *Journal of Integrative Bioinformatics*, vol. 1(1):2, 2004.

[20] G. A. Pavlopoulos, S. I. O'Donoghue, V. P. Satagopam, et al., "Arena3D: visualization of biological networks in 3D," *BMC Syst Biol*, vol. 2, p. 104, 2008.

[21] R. R. Ishiwata, M. S. Morioka, S. Ogishima, et al., "BioCichlid: central dogma-based 3D visualization system of time-course microarray data on a hierarchical biological network," *Bioinformatics*, vol. 25, pp. 543-4, Feb 15 2009.

[22] R. Bourqui, L. Cottret, V. Lacroix, et al., "Metabolic network visualization eliminating node redundance and preserving metabolic pathways," *BMC Syst Biol*, vol. 1, p. 29, 2007.

[23] T. Dwyer, B. Lee, D. Fisher, et al., "A comparison of user-generated and automatic graph layouts," *IEEE Trans Vis Comput Graph*, vol. 15, pp. 961-8, Nov-Dec 2009.

[24] B. Johnson and B. Shneiderman, "Tree-Maps: a space-filling approach to the visualization of hierarchical information structures," in *Proceedings of the 2nd conference on Visualization '91*, San Diego, California, 1991, pp. 284-291.

[25] B. Shneiderman, "Tree visualization with tree-maps: 2-d space-filling approach," *ACM Transactions on Graphics*, vol. 11, pp. 92-99, 1992.

[26] J. Yang, M. O. Ward, E. A. Rundensteiner, et al., "InterRing: a visual interface for navigating and manipulating hierarchies," *Information Visualization*, vol. 2, pp. 16-30, March 2003

[27] G. G. Robertson, J. D. Mackinlay, and S. K. Card, "Cone Trees: animated 3D visualizations of hierarchical information," in *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, New Orleans, Louisiana, United States, 1991, pp. 189-194.

[28] T. Munzner, "Exploring Large Graphs in 3D Hyperbolic Space," *IEEE Computer Graphics and Applications*, vol. 18, pp. 18-23, 1998.

[29] J. Fekete, Wang, D., "Overlaying Graph Links on Treemaps," in *In Information Visualization 2003 Symposium Poster Compendium, IEEE*, 2003, pp. 82-83.

[30] S. Carbon, A. Ireland, C. J. Mungall, et al., "AmiGO: online access to ontology and annotation data," *Bioinformatics*, vol. 25, pp. 288-9, Jan 15 2009.

[31] S. Maere, K. Heymans, and M. Kuiper, "BiNGO: a Cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks," *Bioinformatics*, vol. 21, pp. 3448-9, Aug 15 2005.

[32] P. Saraiya, P. Lee, and C. North, "Visualization of Graphs with Associated Timeseries Data," in *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*: IEEE Computer Society, 2005.

[33] *MetNet database (MetNetDB)* http://metnet.vrac.iastate.edu/MetNet_db.htm,

[34] T. Tekusova and T. Schreck, "Visualizing Time-Dependent Data in Multivariate Hierarchic Plots -Design and Evaluation of an Economic Application," in *Information Visualisation, 2008. IV '08.*, Columbus, OHIO, USA, 2008, pp. 143-150.

[35] J. Ellson, E. R. Gansner, and E. Koutsofios, "Graphviz and dynagraph static and dynamic graph drawing tools," Technical report, AT&T Labs - Research 2003.

[36] E. G. Zoetendal, A. H. Smith, M. A. Sundset, et al., "The BaeSR two-component regulatory system mediates resistance to condensed tannins in Escherichia coli," *Appl Environ Microbiol*, vol. 74, pp. 535-9, Jan 2008.

[37] M. Jia, L. Li, E. Boggess, et al., "Visualizing Multivariate Hierarchic Data Using Enhanced Radial Space-Filling Layout," in *Advances in Visual Computing*. vol. 6453, Springer Berlin / Heidelberg, 2010, pp. 350-360.

[38] M. Jia, S.-Y. Choi, D. Reiners, et al., "MetNetGE: interactive views of biological networks and ontologies," *BMC Bioinformatics*, vol. 11, p. 469, 2010.

[39] A. Katifori, C. Halatsis, G. Lepouras, et al., "Ontology visualization methods a survey," *ACM Computing Surveys*, vol. 39, p. 10, 2007.

[40] A. Katifori, E. Torou, C. Vassilakis, et al., "Selected results of a comparative study of four ontology visualization methods for information retrieval tasks," in *Research Challenges in Information Science, 2008. RCIS 2008. Second International Conference on*, 2008, pp. 133-140.

[41] E. H. Baehrecke, N. Dang, K. Babaria, et al., "Visualization and analysis of microarray and gene ontology data with treemaps," *BMC Bioinformatics*, vol. 5, p. 84, Jan 2004.

[42] J. Stasko, "An evaluation of space-filling information visualizations for depicting hierarchical structures," *International Journal of Human-Computer Studies*, vol. 53, pp. 663-694, 2000.

[43] Gene Ontology Consortium, *GO Slim and Subset Guide*,

http://www.geneontology.org/GO.slims.shtml, 2011.

[44] G. B. Casella, Roger L., *Statistical inference*, 2nd ed., Duxbury,2001.

[45] S. S. W. Shapiro, M. B., "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, pp. 591–611, 1965.

[46] *Shapiro–Wilk test*, http://en.wikipedia.org/wiki/Shapiro-Wilk_test, 2011.

[47] P. N. Johnson-Laird, "Mental models and human reasoning," *Proc Natl Acad Sci U S A*, vol. 107, pp. 18243-50, Oct 2010.

[48] P. N. Johnson-Laird, V. Girotto, and P. Legrenzi, "Reasoning from inconsistency to consistency," *Psychol Rev*, vol. 111, pp. 640-61, Jul 2004.

[49] D. A. Norman, "Some observations on mental models," in *Human-computer interaction*, R. M. Baecker, Ed. Waltham, MA, Morgan Kaufmann Publishers Inc., 1987, pp. 241-244.

[50] S. J. Payne, "Users' Mental Models: The Very Ideas.," *HCI Models, Theories, and Frameworks*, pp. 135-154, 2003.

[51] J. Preece, Rogers, Y., Sharp, H., Benyon, D., Holland, S.,, *Human-Computer Interaction*, Addison-Wesley.,1994.

[52] K. Koffka, "Principles of Gestalt Psychology," New York: Harcourt Brace, 1935.

[53] P. O. Gray, *Psychology*, 6th ed., New York, NY, Worth Publishers,2010.

[54] *The Gestalt Principles*,

http://graphicdesign.spokanefalls.edu/tutorials/process/gestaltprinciples/gestaltprinc.htm,